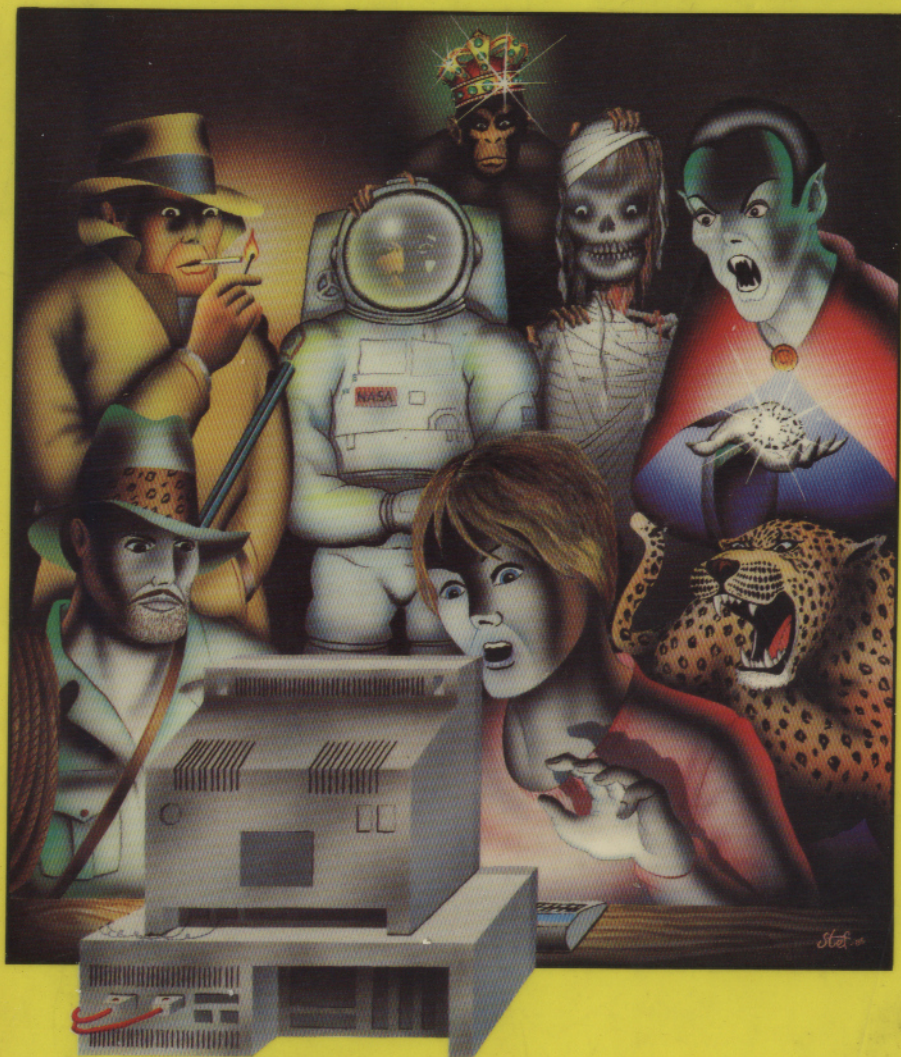


Lav dine egne

COMPUTER EVENTYR

med BASIC



Niels Søndergaard

Borgen

Lav dine egne

Computer- eventyr

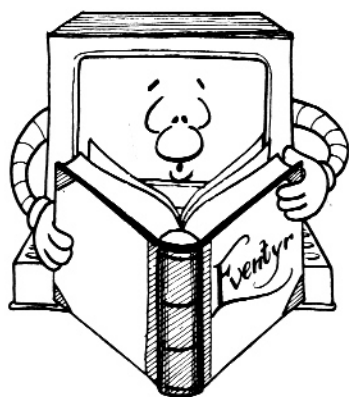
med BASIC

NIELS SØNDERGAARD

Lav dine egne

Computer- eventyr

med BASIC



BORGEN

© 1985 Niels Søndergaard
Omslag og vignetter: Steffen Vedsted
Trykt hos Reproset
ISBN 87-418-7531-1

All rights reserved
Alle rettigheder, herunder
retten til erhvervsmæssig
udnyttelse af bogens programmer,
forbeholdes.

Bogens programmer, herunder en udvidet udgave af »På skattejagt i Amazonlandet«, kan leveres på diskette til visse computere. Nærmere oplysninger om tekniske specifikationer og pris kan fås hos Borgens Forlag, Valbygårdsvej 33, DK-2500 Valby, Danmark.

Indhold

Forord 7

1 · Hvad er eventyrspil? 9

2 · Huset med de ti rum 16

3 · Det mystiske rumskib 54

4 · Nogle videre finesser 98

5 · På skattejagt i Amazonlandet 128

6 · Tilfældigt anbragte lokaliteter 186

7 · Faraos skat 203

8 · Eventyrspil på disketter 231

9 · Mordet i den lukkede boks 235

Stikordsregister 263

Forord

Dette er en bog for begyndere – dog kun hvad eventyrspil angår. En forudsætning er, at man har et i det mindste nødtørftigt kendskab til BASIC – for »Lav dine egne COMPUTEREVENTYR med BASIC» er *ikke* en grundbog i BASIC-programmering.

På den anden side er netop eventyrspil, der opererer med et begrænset antal BASIC-kommandoer og instruktioner, en glimrende øvelse for nybegyndere i programmeringens kunst. Så hvis du har en pæn portion (tål)mod og en BASIC-håndbog, som du kan konsultere undervejs – ja, så køр bare løs. Men kom ikke bagefter og sig, at du ikke er blevet advaret!

Angående de udskrifter, du vil finde i denne bog: programmerne er lavet i Microsoft BASIC, men skulle være ret nemme at konvertere til andre BASIC-dialekter, da de som nævnt er holdt i et så enkelt sprog som overhovedet muligt, ligesom ingen af linierne overstiger 80 anslag (også af somme kaldet »karakterer«) – og i øvrigt er hvert enkelt programmeringsproblem detaljeret forklaret, så enhver kan følge med i, hvad der foregår. Som en speciel service for Commodore-brugere er der hvert sted, hvor Microsoft-versionen afviger fra Commodore-versionen, anført, hvordan man foretager en konvertering (samtlige programmer har været prøvekørt på en Commodore 64 uden problemer). Intet steds i bogen vil du støde på de kommandoer, der adskiller sig mest fra datamat til datamat: grafik- og farvekommandoerne. Denne bog handler nemlig om *tekst*spil. Hvis du vil have kulør på dine eventyr, må du selv lægge den på bagefter.

Spillene i denne bog fylder (renset for overflødige REM-sætninger og i LOAD'et tilstand):

DEMONSTRATIONSSPILLET: ca. 3000 bytes.

DET MYSTISKE RUMSKIB: ca. 16000 bytes.

PÅ SKATTEJAGT I AMAZONLANDET: ca. 37000 bytes.

FARAOS SKAT: ca. 28000 bytes.

MORDET I DEN LUKKEDE BOKS: ca. 27000 bytes.

God fornøjelse!

Niels Søndergaard

Juli 1985

1 · Hvad er eventyrspil?



Der var meget tyst i junglen. Kun cikadernes sang gennemtrængte den trykkende middagshede. Montana Smith stirrede eftertænksomt på den opdagelsesrejsendes skelet og kiggede endnu en gang på den dagbog, han havde fundet ved siden af liget. »Jeg går med en guldnøgle i en kæde om halsen«, stod der . . . men der var ingen guldnøgle at se. Hvor var den? Montana Smith rynkede panden.

Pludselig fik en raslen i underskoven ham til at kigge op. Bloddet stivnede i hans årer. En kæmpestor leopard kom langsomt og truende krybende hen imod ham. Dens hale svirpede fra side til side, og den knurrede dybt nede i brystet. Resolut greb han sin riffel, tog sigte og trykkede af . . . KLIK, sagde det.

»Pokkers! Jeg har ikke noget ammunition!« hviskede han, og sveden, der dagen lang havde piblet frem af hans hud, blev kold . . . kold som is. »Jeg glemte at købe ammunition, da jeg var på handelsstationen! Hvor dum kan man være?«

Leopardens knurren gled over i et brøl, der fik urskovens kroner til at skælve. Den krummede bagbenene sammen til spring . . .

Når vi læser eventyr eller spændende historier, bliver vi revet med, opslugt af handlingen, går så at sige ind i den – og *bliver* helten (eller heltinden!), identificerer os med hans/hendes situation, lever og ånder vor hovedpersons liv – i hvert fald, hvis det er en god historie! Men én ting, vi ikke kan gøre, er at bestemme udfaldet af historien eller dens enkelte bestanddele, bestemme hvad personen skal gøre i en given situation, for det er ifølge sagens natur allerede afgjort og nedfældet på papir af forfatteren. Hvis denne har bestemt, at Montana Smith skal flygte fra leoparden, *flygter* han fra leoparden, og hvis han skal slås ihjel af leoparden, *bliver* han slået ihjel af leoparden – læseren kan intet stille op, men er nødt til at følge den handling, forfatteren har afstukket.

Men tænk nu, hvis vi *kunne* afgøre Montana Smiths skæbne . . . Tænk, hvis vi havde mulighed for at styre hans handlinger og selv bestemme, hvad der skulle ske med ham . . . Tænk, hvis vi *var* Montana Smith:

**DU STAR I EN LYSNING,
HVOR DU SER SKELETTET
AF EN OPDAGELSESREJSENDE.**

**DU KAN SE:
EN DAGBOG**

**DU KAN GÅ:
NORDPA**

**HVAD VIL DU GØRE?
Tage dagbogen**

UDMÆRKET.

(Skærmen ryddes)

HVAD VIL DU GØRE?

Undersøge dagbogen

DER STAR NOGET I DEN.

HVAD VIL DU GØRE?

Læse dagbogen

I DAGBOGEN STAR BL.A.:

'26. AUGUST.

I DAG FANDT JEG EN LILLE

GULDNØGLE, SOM JEG NU

GAR MED I EN KÆDE OM HALSEN.'

HVAD VIL DU GØRE?

Tage guldnøgle

FINDES IKKE HER.

HVAD VIL DU GØRE?

Undersøge stedet

DET BLIVER DU IKKE KLOGERE AF.

PLUDSELIG BLIVER DU ANGREBET

AF EN KÆMPESTOR LEOPARD.

HVAD VIL DU GØRE?

Skyde leoparden

DU HAR IKKE NOGET AMMUNITION.

HVAD VIL DU GØRE?

Gå mod nord

DER ER NOGET, DER TRUER DIG,

OG DU KAN IKKE BARE STILLE

OG ROLIGT GA DIN VEJ.

HVAD VIL DU GØRE?

Status

DU BÆRER FØLGENDE:

EN RIFFEL

EN MESSINGNØGLE

EN KNIV

HVAD VIL DU GØRE?

Brug messingnøglen

DET FAR DU IKKE NOGET UD AF.

HVAD VIL DU GØRE?

Brug kniven

DU GAR FOR RASK TIL VÆRKS,

OG KNIVEN KNÆKKER.

HVAD VIL DU GØRE?

Flygte!!

DU SMIDER ALT, HVAD DU HAR PÅ DIG,

OG FLYGTER OVER HALS OG HOVED

UDEN AT TÆNKE OVER

I HVILKEN RETNING DU LØBER.

Hov, hvad var nu det? Hvor blev alle stemningsbeskrivelserne af? De malende billeder? Sveden, der pibler ud af Montana Smiths porer? Leopardens piskende hale? Dens uhyggelige knurren? Den trykkende middagshede? Savnede du dem? Eller syntes du, det var lige så spændende (eller måske endnu mere spændende) selv at få lov til at lege Montana Smith, der er på jagt efter en skat i Amazonlandets uigennemtrængelige jungle?

Det, du lige læste, var en udskrift af et såkaldt eventyrspil, skrevet til en computer. Det, der var skrevet med store bogsta-

ver, var computerens output, og det, der (for overskuelighedens skyld) stod med små bogstaver, var læserens input. Læseren har med andre ord indflydelse på handlingen og kan få computeren (læs: eventyrfortælleren) til at variere handlingen. Hvis Montana Smith fx havde husket at KØBE AMMUNITION, da han var ved handelsstationen, ville han ikke have haft besvær med at SKYDE LEOPARDEN – og dog . . . for der er ingen, der siger, at han nødvendigvis kunne ramme den. Også dette er en af de ting, der afgøres af den, der programmerer eventyret – og hensigten med denne bog er at lære *dig* at programmere eventyrspil.

Som jeg allerede har antydnet, er den mest markante forskel på eventyr i bøger og så computer-eventyr netop, at *man selv* har en vis indflydelse på handlingen, når man læser et eventyr skrevet til en computer. Når jeg skriver »en vis indflydelse«, mener jeg, at forfatteren af computer-eventyret naturligvis også har lavet en slutning på det . . . men det er *læseren*, der bestemmer, om historien skal ende godt eller skidt . . . det er *læseren*, der bestemmer, hvor længe den skal vare . . . det er *læseren*, der bestemmer, hvad vores helt skal foretage sig, hver gang han får et valg (og det får han – eller hun – hele tiden), og det er derfor, genren har fået navnet eventyrspil – for det *er* spil.

Når jeg overhovedet drager bøger ind i billedet, og når jeg bruger ordet »læseren« i stedet for »spilleren« er det for at understrege, at eventyrspil efter min mening er mere i familie med litteratur end med de såkaldte blip-båt-spil, man normalt forbinde med computere – spil med tegninger og farver, der mest går ud på at teste spillerens reaktionsevne . . . om han når at skyde alle de angribende rumskibe ned, før missilsiloerne er udbombet . . . om osten når at æde alle de små energikugler, før spøgelserne får ram på den. Eventyrspil er først og fremmest *tekstspil*, der stimulerer intellektet, hvorimod blip-båt-spil er grafik- og farvespil, der som regel udelukkende stimulerer brugerens fysiske reaktionsevne. Eventyrspil er i deres reneste form markant forskellige fra blip-båt-spil – og forstå mig ret: jeg har ikke det fjerneste mod blip-båt-spil! Jeg ønsker blot at slå fast, at eventyrspil *er* anderledes.

Men de er så sandelig også forskellige fra de eventyr, vi læser i bøgerne, dels fordi læseren/spilleren som sagt til en vis grad

kan bestemme, hvad der skal ske, dels fordi de af hensyn til computerens begrænsede hukommelseskapacitet kun kan medtage den allernødvendigste tekst. Hvis du er i tvivl om, hvad jeg mener, så prøv endnu en gang at sammenligne det »skrevne« eventyr i begyndelsen af kapitlet med computerudskriften.

Selv om det naturligvis kun er et spørgsmål om tid, før vi får hjemmecomputere med gigantiske hukommelser, der er i stand til at medtage de malende beskrivelser, der er i »romaneventyret«, vil jeg nu tro, at computereventyret stadig vil nøjes med at operere med begrænsede tekstmængder – fordi spilleren er ivrig efter at komme videre og alligevel vil springe alenlange beskrivelser af Montana Smiths svedudsondring og leopardens piskenende hale over, utålmodigt indtaste sin næste kommando og med tilbageholdt åndedræt trykke på returtasten for at se, hvad den djævelske maskine *nu* har fundet på af nederdrægtigheder. Spilleren er så at sige selv med til at skrive historien, både på skærmen og inde i sit hoved – behøver ikke få den uddybet yderligere. Det hele fremtræder mere som en dyst, en quiz, end som en litterær oplevelse, og eventyrspil *er*, slægtskabet til trods, da heller ikke litteratur, men en selvstændig kunstart, der er i riven-
de udvikling – på indeværende tidspunkt hovedsagelig i USA og England – og jeg vil tro, at den allerede har taget de ting, den har brug for fra litteraturen, ligesom den har lånt mange ingredienser fra de i Danmark forholdsvis ukendte bræt-rollespil som »Dungeons and Dragons« og på det sidste er begyndt at inddrage blip-båt-spillenes grafiske elementer.

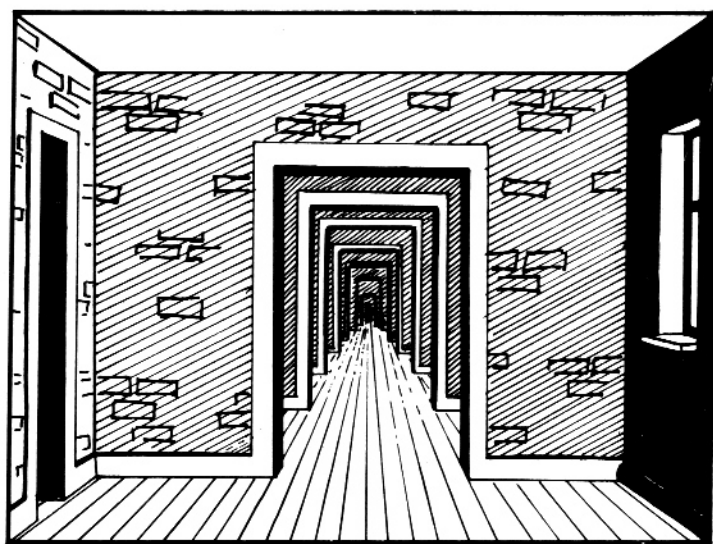
Lad os gå et par år tilbage og se på, hvordan det hele egentlig begyndte.

Midt i halvfjerdserne skabte to amerikanske programmører, Willie Crowther og Don Woods, historiens første eventyrspil på Stanford-universitetets Mainframe-computer. De kaldte det »Colossal Caves«, men da det fyldte cirka 300 K i datamatens hukommelse, og da der i øvrigt ikke på dette tidspunkt var noget, der hed hjemmecomputere, var der ikke mange muligheder for at få det bragt ud til almindelige mennesker. Det var først med hjemmecomputer-eksplosionen, at eventyrspillet blev hvermands eje, og en stor fortjeneste for dette har amerikaneren Scott Adams, der regnes for at være den første, der begyndte at

skrive eventyrspil til små computere. Scott Adams' spil er stadig nogle af de bedste, der nogen sinde er lavet, og han er i dag millionær og har sit eget firma, der udelukkende beskæftiger sig med at producere eventyrspil. Genren er i dag som sagt uhyre populær i USA og England, og populariteten synes at være stigende. I skrivende stund findes der ikke særlig mange eventyrspil på det danske marked, og af disse er kun ganske få på dansk, men jeg tror og håber, at dette vil ændre sig.

Og lad os så se at komme i gang!

2 · Huset med de ti rum



Alfa og omega i et eventyr – i enhver historie – er naturligvis *handlingen*. Hvis man ikke har en handling, er der ikke megen idé i at begynde at skrive et eventyr. Handlingen kan være ganske enkel: Hovedpersonen bevæger sig fra Sted A til Sted Z og oplever ting undervejs. Eller hovedpersonen *bliver* på Sted A – der er sådan set ikke noget i vejen for at skrive en historie, der foregår på samme sted... men de fleste eventyrspil opererer med forskellige lokaliteter, som spilleren skal udforske – og undervejs finde ting, han kan samle op og bruge til at overvinde forhindringer med, indtil han til sidst finder skatten, redder verden, får prinsessen – eller hvad det nu er, han forventes at slutte af med.

Men lad os indtil videre glemme handlingen og kigge på de ingredienser, der skal til, for at vi overhovedet kan begynde at

tænke på at konstruere en handling. Man må krybe, før man kan gå – og det gælder i allerhøjeste grad også for eventyrspilforfattere. Men i dette tilfælde er det heldigvis ikke så svært at lære at krybe – i hvert fald ikke, hvis man har kendskab til de mest elementære BASIC-kommandoer. Så, endnu en gang: hvis dette er din første indføring i computer-programmeringens fascinerende verden, så læs en grundbog i BASIC først – eller i hvert fald jævnsides med denne. Hvis jeg skal være helt ærlig, lærte jeg selv at programmere netop ved at kaste mig ud i halsbrækkende opgaver, jeg ikke havde mange forudsætninger for at løse ... og blev meget, meget klogere undervejs. En sådan fremgangsmåde koster mange fejltagelser og ærgrelser – men til gengæld lærer man noget af den. Gør, hvad du vil, kære nybegynder – men som jeg allerede har sagt: kom ikke bagefter og sig, at du ikke er blevet advaret!

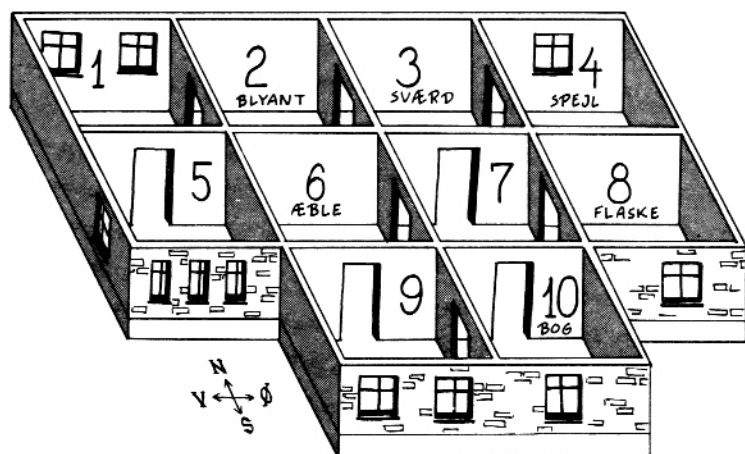
Efter denne lille skolemesterprædiken var det måske på sin plads at komme i gang med de tørre fakta. Vi fortsætter, hvor vi slap – med *de elementære byggesten*.

Vi kan opstille en grov model over, hvilke ingredienser vi får brug for, når vi skal skrive et eventyr. Vi skal naturligvis have en hovedperson – men det er spilleren, som findes i forvejen, så det problem *er* løst. Vi kan også indføje bipersoner – hvis vi fx laver et spil over eventyret om Rødhætte, skal vi foruden Rødhætte (lad os sige, at *hun* er spilleren) have ulven, bedstemoderen og jægeren med. Men lad os nu ikke tage for store mundfulde her i begyndelsen. Lad os nøjes med én person – som altså er den »DU«, som computeren henvender sig til (dvs. spilleren).

Så skal vi bruge et »landskab«, spilleren kan bevæge sig rundt i. Når jeg skriver »landskab« i gåseøjne, er det fordi et eventyrspil absolut ikke behøver foregå ude i det fri – mange af de bedste foregår faktisk i lukkede omgivelser – skumle borge, kringledede huler, forladte rumskibe ... så lad os i stedet sige, at vi behøver en række lokaliteter, vi kan bevæge os rundt imellem. Lad os forestille os, at vi befinder os i et hus uden yderdøre – et lukket og forseglet hus med ti rum. I disse ti rum vil vi lægge forskellige ting, vi kan samle op (og lægge fra os igen). Eftersom vores demonstrationsmodel ikke har nogen handling, men udelukkende tjener til at vise, hvordan man får sin person til at bevæge sig fra

rum til rum og tage eller lægge forskellige ting, kan det være ret ligegyldigt, hvad det er for ting. Lad os tage seks helt tilfældige og vidt forskellige objekter: en flaske, et æble, en bog, en blyant, et sværd og et spejl.

Så er vi faktisk parat. Vores opgave er at konstruere et hus med ti rum, hvori der ligger seks genstande. Lad os for over-skuelighedens skyld begynde med at tegne et kort:



Som du kan se, er de ti rum nummereret fra 1-10. De er forbundet med dobbelpile, der markerer dørene i huset. Vi kan altså notere os, at vi også har brug for *døre*, der forbinder lokaliteterne med hinanden. De markerer, i hvilken retning vi kan gå, og manglen på døre markerer så naturligvis i hvilken retning, vi *ikke* kan gå. Står vi fx i Rum nr. 3, kan vi gå mod syd, øst og vest, men ikke mod nord. Står vi i Rum nr. 8, kan vi kun gå mod vest. De seks genstande er tilfældigt anbragt i Rum nr. 2, 3, 4, 6, 8 og 10.

Men stop! Hvordan kommer dette til at se ud på skærmen? Det er jo tekstspil, vi snakker om, så hvordan kan vi »se« sværdet i Rum nr. 3? Hvordan kan vi »se« de tre udgange mod syd, øst og vest? Og hvordan i alverden kan vi *flytte* sværdet fra Rum nr. 3 til fx Rum nr. 7?

Okay, lad os foregribe begivenhedernes gang en smule og vise en skærmudskrift fra vores færdige demonstrationsspil:

DU STAR I RUM NR. 3

DU KAN SE:

ET SVÆRD

DU KAN GA:

SYDPA

ØSTPA

VESTPA

HVAD VIL DU GØRE?

Tage sværd

UDMÆRKET.

(Skærmen ryddes)

DU STAR I RUM NR. 3

DU KAN GA:

SYDPA

ØSTPA

VESTPA

HVAD VIL DU GØRE?

Lige et øjeblik! Hvor blev sværdet af? Du tog det, og det forsvandt fra skærmen – men hvordan kan du se, at du har det?

HVAD VIL DU GØRE?

Status

DU BÆRER FØLGENDE:

ET SVÆRD

Okay, så er du klar over *det*: Hvis du skriver »Status«, får du at vide, hvad du bærer rundt på – og hvis du ikke bærer rundt på noget, får du beskeden

DU BÆRER FØLGENDE:

INGENTING

Så vidt, så godt. Lad os prøve at gå mod syd:

DU STAR I RUM NR. 3

DU KAN GÅ:

SYDPA

ØSTPA

VESTPA

HVAD VIL DU GØRE?

Gå sydpå

(Skærmen ryddes)

DU STAR I RUM NR. 7

DU KAN GÅ:

NORDPA

SYDPA

ØSTPA

VESTPA

HVAD VIL DU GØRE?

Lægge sværd

UDMÆRKET.

(Skærmen ryddes)

DU STAR I RUM NR. 7

DU KAN SE:

ET SVÆRD

DU KAN GA:

NORDPA

SYDPA

ØSTPA

VESTPA

HVAD VIL DU GØRE?

Status

DU BÆRER FØLGENDE:

INGENTING

Som du kan se, har vi altså skabt en illusion af, at du har taget sværdet i Rum nr. 3 og er gået gennem døren til Rum nr. 7, hvor du har lagt sværdet fra dig igen. Disse to elementære ting: at gå fra ét sted til et andet og at tage eller lægge en genstand, er hovedingredienserne i et eventyrspil. Har vi først lært at beherske dem, er resten – som du siden skal se – forholdsvis enkelt.

Men hvordan gør vi det? Hvordan laver vi et program, der kan få os til at trave rundt i et hus eller en hule eller en skov og samle sværd, æbler, flasker og alt muligt andet op?

Lad os gå systematisk til værks og begynde med lokaliteterne.

Dem skal vi bruge ti af:

RUM NR. 1

RUM NR. 2

RUM NR. 3

etc. op til

RUM NR. 10

Vi har med andre ord ti *DATA*, vi gerne vil putte ind i vores computer. Og hvordan gør vi så det? Jo, for at lette arbejdet for vores computer anbringer vi dem i et DATAsæt – et sæt, der består af ti indekserede variabler: L\$(1), L\$(2), L\$(3) etc. op til L\$(10). Bemærk, at vi ved hjælp af dollartegnet markerer, at de er strengvariabler, da det jo her er ord og ikke tal, vi opererer med:

L\$(1) = "RUM NR. 1"

L\$(2) = "RUM NR. 2"

L\$(3) = "RUM NR. 3"

etc. op til

L\$(10) = "RUM NR. 10"

Men vent! Ville det ikke være nemmere at arrangere dem i en DATA-sætning og dermed spare både plads og gåseøjne? Kunne vi ikke skrive

DATA RUM NR. 1, RUM NR. 2, RUM NR. 3

etc.?

Jo, slet ikke nogen dum idé – bortset fra, at der er to gode grunde til at skrive netop lokalitetsvariablerne op hver for sig: L\$(1)=»RUM NR. 1« etc. Disse grunde vil vi straks vende tilbage til, men ideen er som sagt slet ikke dum, så lad os anvende den på de *øvrige* data.

Men lad os først slå en ting fast: det, vi indtil videre har fortalt computeren, er *hvad den skal skrive ud på skærmen*, så spilleren kan læse det: RUM NR. 1 for eksempel. Men computere opererer med tal, og ord er noget, de ikke bryder sig om og insisterer på at få serveret som strengvariabler, som de så i deres indre kan bruge som tal. Indholdet af disse strengvariabler er de bedøvede ligeglade med. Indholdet (ordlyden) er udelukkende beregnet på at kunne læses af spilleren, der ikke kan forventes at huske, at en bestemt variabel betyder fx RUM NR. 1. Så for at fortælle *computeren*, i hvilket rum vi befinder os, bruger vi en LOkalitetsvariabel, som vi kalder LO. Og hvis vi giver computeren instruktionen

IF LO=A THEN PRINT L\$(A)

eller mere konkret:

IF LO=1 THEN PRINT L\$(1)

vil computeren, dersom LO er lig 1, på skærmen skrive:

RUM NR. 1

Lad dig ikke forvirre, hvis du ikke umiddelbart forstår dette. Det bliver udførligt forklaret om lidt. Men lad os gå videre: vi skal også sætte vores computer i stand til at skrive seks navngivne genstande på skærmen: en blyant, et sværd, et spejl, et æble, en flaske og en bog. Lad os putte dem ind i maskinen ved hjælp af en DATA-sætning:

```
DATA EN BLYANT,ET SVÆRD,ET SPEJL,ET ÆBLE,EN  
FLASKE,EN BOG
```

Godt . . . men det er naturligvis ikke nok. Vi mangler fx et linie-nummer, der kan fortælle computeren, at dette er en del af et program og ikke skal udføres i *direct mode*. Dette linienummer er naturligvis afhængigt af, *hvor* i programmet DATA-sætningen optræder. Lad os give det fx linienummeret 30:

```
30 DATA EN BLYANT,ET SVÆRD,ET SPEJL,ET ÆBLE,EN  
FLASKE, EN BOG
```

Det sæt, der rummer disse DATA, kalder vi G\$() – G'et står for genstande. Og vi skal huske at DIMensionere det, før vi tager det i brug – dvs. sætte plads af til det i hukommelsen. Selv om sættet er på mindre end ti variabler, som computeren sådan set godt kan hitte rede i uden en DIM-sætning, er det en god idé at DIMensionere det for at spare hukommelsesplads: så bruger computeren nemlig kun den plads, der skal til for at rumme de seks DATA. Og hvis sættet er på mere end ti variabler, *skal* det under alle omstændigheder DIMensioneres. Så DIMensionér *altid* dine sæt. Vi DIMensionerer vores genstands-sæt på følgende enkle måde:

```
10 DIM G$(6)
```

hvor 10 er linienummeret, der fortæller computeren, at dette er *indirect mode* – altså et program.

Men det er stadig ikke nok. Computeren ved nu, at DATA-sættet G\$() indeholder seks variabler. Men selv om vi har givet den linie 30, kan den ikke læse den endnu. På dette punkt er computeren ret dum, og vi er nødt til at give den ordre til at læse DATA-sætningen. Det gør vi med en READ-instruktion:

```
20 FOR I=1 TO 6:READ G$(I):NEXT I
```

Instruktionen betyder ganske enkelt, at computeren indlæser de første seks DATA, den støder på, og i sin hukommelse noterer at:

```
G$(1)="EN BLYANT"  
G$(2)="ET SVÆRD"  
G$(3)="ET SPEJL"  
G$(4)="ET TÆBLE"  
G$(5)="EN FLASKE"  
G$(6)="EN BOG"
```

Når vi skal indlæse DATA i maskinen, sker det altså i tre etaper:

```
1 DIM X$(A)
```

hvor X\$() er navnet på (streng)sættet og A er det antal DATA, sættet skal indeholde.

```
2 FOR I=1 TO A:READ X$(A):NEXT I
```

hvor computeren får ordre til at læse (READ'e) DATA-sætningen, der sådan set kan stå hvor som helst i programmet, men som i dette eksempel kommer nu:

```
3 DATA AAA,BBB,CCC
```

etc. A gange.

Vores genstands-sæt vil således kunne se sådan ud:

```
10 DIM G$(6)
20 FOR I=1 TO 6:READ G$(I):NEXT I
30 DATA EN BLYANT,ET SVÆRD,ET SPEJL,ET ÆBLE,EN
    FLASKE,EN BOG
```

Og hvis du tilføjer en linie 40 for at afprøve dette miniprogram (som ikke må forveksles med selve demonstrationsprogrammet):

```
40 PRINT G$(3)
```

vil computeren på skærmen skrive:

```
ET SPEJL
```

Prøv at erstatte G\$(3) i linie 40 med fx G\$(4) eller G\$(6) og se, hvad der sker.

Hvad skal vi mere bruge? Jo, computeren skal kunne fortælle spilleren, hvilke retninger (eller døre, om du vil), han kan operere med. Dem er der i vores eksempel fire af: Han skal kunne gå NORDPÅ, SYDPÅ, ØSTPÅ og VESTPÅ. Der er ikke noget i vejen

for at lave et eventyr, hvor man også kan gå OPAD, NEDAD, SYDØSTPÅ eller NORDVESTPÅ, men lad os ikke gabe over mere, end vi kan overkomme. Lad os nøjes med de fire verdenshjørner. Også dem arrangerer vi et eller andet sted i programmet i en DATA-sætning:

DATA NORDPA, SYDPA, ØSTPA, VESTPA

og vi husker at DIMensionere dem med instruktionen

DIM V\$(4)

hvor V\$() står for verdenshjørner, hvorefter vi indlæser (READ'er) disse DATA i computeren:

FOR I=1 TO 4:READ V\$(I):NEXT I

Nu ved vores computer, hvilke genstandsbeskrivelser og verdenshjørneangivelser, den har mulighed for at skrive på skærmen. Men som jeg før nævnte, er det jo ikke kun på skærmen, det foregår. Det, der står på skærmen, er kun en skriftligt formuleret status over forskellige variablers værdi i computerens hukommelse, som vi jo ikke kan se. Værdien af variabelen LO, som jeg nævnte før, fortæller fx computeren, hvilket rum, du befinder dig i, hvorefter den med et skriftligt udsagn (fx RUM NR. 10) fortæller *dig* det. Eftersom du kun kan befinde dig i ét rum ad gangen, behøver vi kun én variabel med navnet LO – men hvad med alle de genstande, du kan slæbe rundt på (i dette tilfælde indtil seks)? Hvordan holder den rede på dem? Hvordan kan den vide, om sværdet fx ligger i Rum nr. 6 eller Rum nr. 8? Hvordan kan den vide, om du har det på dig?

For at fortælle den dette opretter vi endnu et DATAsæt – og

eftersom det kun er beregnet til computerens interne brug, kan vi her nøjes med at bruge tal. Lad os kalde det $G()$; det gør det nemmere for os at huske, at det hænger sammen med streng-sættet $G\$()$, som vi allerede har oprettet:

DATA 2,3,4,6,8,10

Vi husker naturligvis at DIMensionere og READ'e sættet $G()$, hvorefter computeren ved, at Genstand nr. 1 (blyanten) befinder sig i Rum nr. 2, Genstand nr. 2 (sværdet) befinder sig i Rum nr. 3, Genstand nr. 3 (spejlet) befinder sig i Rum nr. 4, Genstand nr. 4 (æblet) befinder sig i Rum nr. 6, Genstand nr. 5 (flasken) befinder sig i Rum nr. 8, og Genstand nr. 6 (bogen) befinder sig i Rum nr. 10.

Hvis du så *tager* genstanden (fx bogen), giver vi den værdien -1, hvorefter computeren (der et andet sted i programmet vil få dette at vide) er klar over, at du har den på dig. Og hvis du igen lægger bogen fra dig i fx Rum nr. 3, giver vi denne alias variabelen $G(6)$ værdien 3, hvorefter computeren vil være klar over, at bogen er i Rum nr. 3 – at $G(6)=3$.

Hvis du ikke forstår dette, så læs det en gang til. Hvis du stadig ikke forstår det, så om igen! Det er vigtigt, at du er inde i den grundlæggende mekanik, før du går videre.

Lad os vende tilbage til »dørene«. Vi har fortalt vores computer, hvad den skal meddele spilleren med hensyn til disse (NORDPÅ, SYDPÅ, ØSTPÅ, VESTPÅ), men ikke *hvor*. Hvordan viser vi den fx, at der kun er én udgang fra Rum nr. 4, og at denne udgang vender mod vest? Igen bruger vi et DATAsæt, men denne gang giver vi det *to* DIMensioner:

DIM L (10,4)

For at forstå ovenstående instruktion skal vi huske, at der er 10 rum (repræsenteret af 10-tallet), der hver har fire vægge (eller

verdenshjørner – repræsenteret af 4-tallet). I virkeligheden har vi altså her brug for at operere med 10 gange 4 variabler – 40 variabler. Når vi stiller dem op på denne måde i stedet for at skrive DIM L(40) er det, fordi det er nemmere for computeren at operere med dem, hvis de bliver knyttet til hvert sit rum – og i øvrigt også nemmere for os at overskue dem, når vi laver eller læser programmet. Måske undrede det dig før, at jeg ikke ville tæste DATA-sættet L\$() ind i en DATA-sætning, men foretrak at skrive

```
L$(1)="RUM NR. 1"
```

```
L$(2)="RUM NR. 2"
```

```
L$(3)="RUM NR. 3"
```

etc.

Måske vil det også undre dig, at jeg ikke skriver de 40 verdenshjørne-variabler, som computeren skal operere med internt, i en DATA-sætning som denne:

```
DATA 0,5,2,0,0,0,3,1,0,7,4,2,0,0,0,3,1,0,0,0,0,9,7,  
0,3,10,8,6,0,0,0,7,6,0,10,0,7,0,0,9
```

men i stedet arrangerer dem i sætninger à fire:

```
DATA 0,5,2,0
```

```
DATA 0,0,3,1
```

```
DATA 0,7,4,2
```

etc.

Det kræver lidt mere computerhukommelse, indrømmet – men hvis vi nu kombinerer dem med vores data for lokalitetsbeskrivelser, vil du sikkert kunne se en mening med galskaben:

L\$(1)="RUM NR. 1"

DATA 0,5,2,0

L\$(2)="RUM NR. 2"

DATA 0,0,3,1

L\$(3)="RUM NR. 3"

DATA 0,7,4,2

etc.

På denne måde vil vi – når vi skriver eller prøver at tyde programmet – have meget nemmere ved at overskue, hvilke verdenshjørne-DATA, der hører til en bestemt lokalitet. I dette tilfælde har vi godt nok ikke brug for at kunne finde numrene på lokalitetsbeskrivelserne, da de jo er nummereret i sig selv: RUM NR. 7 fx, men forestil dig, at der i stedet for RUM NR. 7 stod EN DUNKEL HULE. Så ville det være svært at se, at den dunkle hule var Lokalitetsbeskrivelse nr. 7, hvis den var anbragt i en DATA-sætning:

DATA EN DUNKEL HULE

hvorimod det hele er langt nemmere at overskue, hvis den bliver anført som:

L\$(7)="EN DUNKEL HULE"

Når jeg lægger så stor vægt på, at programmet bliver til at overskue, er det ikke for at glæde spilleren – som sandsynligvis aldrig

kommer til at se programmet LISTet, men for at hjælpe mig selv (og dig) under konstruktionen af spillet. Hvis man af den ene eller den anden grund kommer til at lave sit kort om under opbygningen af spillet (og det gør man masser af gange, hvis man ikke har det hele fuldstændig klarlagt fra begyndelsen – og så skal der nok alligevel dukke et par nye ideer op undervejs!), er det en Guds velsignelse at kunne finde rundt i sit program. Men i øvrigt er der også en anden, meget god grund til at stille lokalitetsbeskrivelserne op på den af mig anførte måde – en grund jeg skal komme nærmere ind på senere. Lad os indtil videre sige, at dersom du alligevel ønsker at putte dine lokalitetsbeskrivelser ind i DATA-sætninger, så skal du naturligvis gøre det. Jeg har selv gjort det i det næste spil, »Det mystiske rumskib«, hvor der næsten ikke *er* nogen lokalitetsbeskrivelser og dermed ikke så meget at overskue. Men i større spil *letter* det overskueligheden at definere hver lokalitetsbeskrivelsesvariabel for sig, og så er der som sagt en anden god grund til at gøre det. Hvis du brænder af nysgerrighed efter at vide, hvad det er for en anden god grund, så kan du jo prøve at smugkigge i afsnittet »Skærmpresentation« i Kapitel 4.

Og lad os så se at komme videre.

Lad os kigge på de DATA, vi har opført i forbindelse med Rum nr. 1: 0,5,2,0.

Vi har allerede angivet vores fire verdenshjørner i følgende rækkefølge: nord, syd, øst, vest. Vi kunne også sige, at nord er Verdenshjørne nummer 1, syd Verdenshjørne nummer 2, øst Verdenshjørne nummer 3 og vest Verdenshjørne nummer 4. Når der i vores L()-DATA for Rum nr. 1 (0,5,2,0) på position nr. 1 står et nul, fortæller det med andre ord computeren, at nord=0, ligesom de øvrige tal fortæller den at syd=5, øst=2 og vest=0. Det første nul viser den, at der ikke *er* nogen dør – at man ikke kan gå mod nord, 5-tallet viser den, at man, dersom man går mod syd, kommer til Rum nr. 5 (se kortet!), 2-tallet, at der er fri passage til Rum nr. 2 mod øst, og det sidste nul, at der er lukket for gennemgang mod vest. Og computeren vil på skærmen fortælle dig, at

DU KAN GA:

SYDPA

ØSTPA

Men ikke nok med det. Den ved også nu, at hvis du går mod syd, kommer du til Rum nr. 5, og hvis du går mod øst, kommer du til Rum nr. 2. Og hvis du prøver at gå mod nord eller vest (det vil sige: hvis computeren støder på et nul), vil du (når du kører det færdige program) få beskeden:

DU KAN IKKE GA DEN VEJ.

Lad os nu prøve at kigge på første del af vores demonstrations-program:

```
10 REM DEMONSTRATIONSPROGRAM TIL EVENTYRSPIL *****
20 REM AL=ANTALLET AF LOKALITETER * AN=ANTALLET AF
   NAVNEORD *****
30 REM AU=ANTALLET AF UDSAGNSORD *****
40 REM GL=ANTALLET AF GENSTANDE, DER BEFINDER SIG PÅ
   EN LOKALITET *****
50 AL=10:AN=10:AU=3:GL=6
60 REM VERDENSHJØRNER=NORDPA, SYDPA, ØSTPA, VESTPA
   *****
70 REM NAVNEORD=FLASKE, ÆBLE, BOG, BLYANT, SVÆRD,
   SPEJL *****
80 REM UDSAGNSORD = GA,TAGE,LÆGGE *****
90 REM L$( )=BESKRIVELSE AF LOKALITET * L( )=DATA FOR
   DØRE *****
100 REM V$( )=VERDENSHJØRNER (TIL BRUG I MENUEN) *
   G( )=GENSTANDENS LOKALITET *****
110 REM G$( )=BESKRIVELSE AF GENSTAND * U$( )=DATA FOR
   UDSAGNSORD *****
```

```

120 REM N$( )=DATA FOR NAVNEORD *****
130 DIM L$(AL),L(AL,4),V$(4),G(GL),G$(GL),U$(AU),
    N$(AN)
140 FOR I=1 TO AL:FOR J=1 TO 4:READ L(I,J):
    NEXT J:NEXT I
150 FOR I=1 TO 4:READ V$(I):NEXT I
160 FOR I=1 TO GL:READ G(I):NEXT I
170 FOR I=1 TO GL:READ G$(I):NEXT I
180 FOR I=1 TO AU:READ U$(I):NEXT I
190 FOR I=1 TO AN:READ N$(I):NEXT I
200 L$(1)= "RUM NR. 1"
210 DATA 0,5,2,0
220 L$(2)= "RUM NR. 2"
230 DATA 0,0,3,1
240 L$(3)= "RUM NR. 3"
250 DATA 0,7,4,2
260 L$(4)= "RUM NR. 4"
270 DATA 0,0,0,3
280 L$(5)= "RUM NR. 5"
290 DATA 1,0,0,0
300 L$(6)= "RUM NR. 6"
310 DATA 0,9,7,0
320 L$(7)= "RUM NR. 7"
330 DATA 3,10,8,6
340 L$(8)= "RUM NR. 8"
350 DATA 0,0,0,7
360 L$(9)= "RUM NR. 9"
370 DATA 6,0,10,0
380 L$(10)= "RUM NR. 10"
390 DATA 7,0,0,9
400 DATA NORDPA,SYDPA,ØSTPA,VESTPA
410 DATA 2,3,4,6,8,10
420 DATA EN BLYANT,ET SVÆRD,ET SPEJL,ET ÆBLE,EN
    FLASKE,EN BOG
430 DATA GA,TAG,LÆG

```

Før jeg begynder at gennemgå denne del af programmet (bemærk venligst, at det ikke er *hele* programmet), vil jeg bede dig bide mærke i et par ting:

Intervaller mellem linierne er 10. Sørg altid for at lave et sådant interval (gerne større), så du får plads til ekstra linier, når du har brug for det (og det har du som regel altid). Hvis du har fx en Commodore 64, vil den acceptere et program med linienumre op til 65536, så hvis det er på under 600 linier, kan du med sindsro lave et interval på 100. Det sparer dig for at skrive en masse om, hvis du pludselig står og mangler plads til nogle tilføjelser, der absolut skal klemmes ind på et bestemt sted. Visse maskiner har en RENUM-kommando, der omnummererer hele molevitten, når man beder dem om det – men altså desværre ikke Commodore. Så kan du til gengæld glæde dig over, at du ikke har en Spectrum-computer, der ikke vil acceptere programmer med højere linienummer end 9999 – medmindre du *har* en Spectrum; men så er du sikkert i forvejen klar over problemet.

Nå, men lad os gennemgå ovenstående program-sektion linie for linie:

10-40: Udelukkende REM-linier, der er til benefice for os og ikke for computeren. Hver gang, jeg laver en REM-linie, plejer jeg at afslutte den med nogle asterisker (stjerner: *****). Så er linierne nemmere at finde og slette senere, hvis man ønsker at skille sig af med dem for at spare den hukommelsesplads, de optager. Linierne 10-40 forklarer de 4 variabler, jeg har anført i linie 50, der altså er vores første instruktion til computeren, og hvor jeg fortæller den, hvor mange lokaliteter, navneord, udsagnsord og genstande, vi skal operere med i spillet. Når jeg allerede nu vælger at operere med variabler i stedet for tal, er det fordi det gør det hele mere overskueligt – og fordi vi bagefter, hvis vi ønsker at forandre antallet af en af disse kategorier, kun behøver at foretage rettelser dette ene sted i stedet for at skulle gøre det i hele programmet.

50: Se ovenfor.

60-120: Flere REM-linier. I linie 70 kan du se, at de seks objekter, som jeg lige før kaldte »genstande«, nu bliver kaldt »navneord«. De er naturligvis stadig »genstande«, men vil også blive behandlet som »navneord«. Forklaring følger straks – og husk, at dette kun er en REM-linie, der ikke går i pinlige detaljer.

Hvad er så alt dette for noget med »navneord« og »udsagnsord«?! Hvad skal vi bruge det til?

Det skal vi bruge, når vi giver computeren input – fortæller den, hvad vi vil gøre, når den spørger os:

HVAD VIL DU GØRE?

Når det sker, vil vi give computeren en kommando, der består af to ord – et udsagnsord og et navneord, fx TAGE BOG eller LÆGGE ÆBLE eller GÅ NORDPÅ (i sidste eksempel bilder vi maskinen, der tror på alt, hvad vi siger til den, ind at NORDPÅ



er et navneord). Men selve denne proces kommer vi til senere – på indeværende tidspunkt behøver du blot vide, at vi også kommer til at operere med yderligere to DATAsæt, et der identificerer NAVNEORD, og et der identificerer UDSAGNSORD. Gå roligt videre, selv om du studser. Alt vil komme for en dag.

130: Her DIMensionerer vi de DATAsæt, vi får brug for i spillet. Du kan i de ovenstående REM-sætninger se, præcis hvad hvert sæt dækker. Hvis du ikke kan huske, hvad variablerne AL, GL, AU og AN står for, så kig endnu en gang på linierne 10-50.

140: Her indlæser (READ'er) vi vores DATA for døre (de retningsangivelser, computeren skal have til internt brug. De, der skal anføres på skærmen, kommer i næste linie). Vores DATA står længere nede i programmet (linie 210-390 – dog kun hver anden af disse linier). Eftersom sættet har *to* DIMensioner, skal vi her bruge to løkker (styret af variablerne I og J – og husk at skrive NEXT J før NEXT I, hvis du vil undgå problemer).

Apropos indlæsning: vi behøver ikke indlæse sættet L\$() – det med lokalitetsbeskrivelserne – da vi jo ikke har medtaget det i en DATA-sætning, men anført hver variabel for sig (linie 200-380 – hver anden af disse linier) af årsager, som jeg allerede har forklaret. Så husk: hvis der ikke er nogen DATA-sætning, skal vi heller ikke have nogen READ-sætning.

150: Her indlæser vi vores DATA for verdenshjørner (dem, computeren skal kunne skrive på skærmen – i modsætning til dem, den skal bruge til internt brug. De står i forrige linie). Som du kan se, er der fire af dem, og da vi skal kunne læse dem på skærmen som ord, er disse DATA strengvariabler (de står i linie 400).

Men vent! Endnu en gang: Vent! Hvordan kan maskinen »se«, hvilke DATA, der hører til hvilket sæt? Hvornår ved den, hvornår det ene DATAsæt hører op, og det andet begynder?

Det ved den sådan set ikke. Den tæller sig frem. I linie 140 har vi fortalt den, at den skal indlæse 40 DATA, og det gør den så. Den tager de første 40 tal-DATA, den støder på, og når vi derefter i linie 160 giver den ordre til at indlæse endnu 6 tal-DATA, tager den de næste 6, den støder på. Så hvis du har kludret i det

og kun skrevet 39 L()-DATA, vil den bare gå videre (dum som den er) til de næste tal-DATA og tro, at 2-tallet i linie 410 er den fyrretyvende variabel i sættet L(). Dette vil naturligvis i sidste ende resultere i en fejlmeddelelse (OUT OF DATA), medmindre du senere har begået endnu en fejl og i en anden linie skrevet en variabel for meget – så pas på! Sørg omhyggeligt for, at det antal DATA, du kræver indlæst, nøje svarer til det antal, du har anført – ellers vil du få et pokkers mas med at finde fejlen bagefter!

160: Her indlæser vi vores DATA for, hvor genstandene befinder sig (husk, at GL betyder »Genstande, der befinder sig på en lokalitet«). Disse DATA står i linie 410.

170: Vi indlæser vores DATA for, hvad computeren skal skrive ud på skærmen, når den vil fortælle os om genstande. Det gør den med ord, og derfor har vi brug for et strengvariabelsæt. Det ser du i linie 420.

180: Her indlæser vi DATA for de udsagnsord (verber), vi gerne vil have computeren til at forstå, når vi giver den besked på, hvad vi vil gøre. Gå videre. Nærmere forklaring følger.

190: Her indlæser vi vores DATA for de navneord (substantiver), computeren skal forstå. Detaljeret forklaring følger i analysen af linie 440.

200-390: DATA for lokalitetsbeskrivelser og de døre, der knytter sig til dem. Se forklaring ovenfor.

400: DATA for de verdenshjørner, der skal skrives ud på skærmen (dem vi indlæste i linie 150).

410: DATA for, hvor genstandene befinder sig (vi indlæste dem i linie 160). De er til computerens interne brug.

420: DATA for de genstande, der skal skrives ud på skærmen (vi indlæste dem i linie 170).

430: DATA for de udsagnsord, vi putter ind i computeren. En nærmere forklaring på, hvorfor ingen af dem er på mere end tre bogstaver, følger i analysen af næste linie.

440: DATA for de navneord, vi putter ind i computeren, når den spørger os, hvad vi vil gøre. Hvis du undrer dig over, hvorfor der står BLY og ikke BLYANT og SVÆ og ikke SVÆRD etc., er forklaringen den, at vi sparer hukommelsesplads ved kun at anføre de første tre bogstaver i hvert ord (vi sørger senere for, at computeren ikke behøver mere end tre bogstaver for at genkende ordet). Du kan sikkert godt se det praktiske i at undgå at operere med strenge som HØJSPÆNDINGSKABELFORLÆNGERLEDNING og FOLKETINGSMØDESALSGULVTÆPPERENSEMASKINE og i stedet nøjes med at skrive HØJ og FOL (dette vil blive forklaret meget mere detaljeret i gennemgangen af tredje program-sektion). Og hvis du undrer dig over, at nordpå (NOR), sydpå (SYD), østpå (ØST) og vestpå (VES) er medtaget som »navneord«, så er forklaringen den, at vi bilder computeren ind, at de *er* navneord – så passer de nemlig som hånd i handske til udsagnsordet GÅ . . . men mere om dette senere.

450: Her fortæller vi computeren at LO=3. Som du måske husker, er LO (LOkalitet) den variabel, der i maskinens indre viser, hvilket rum vi befinder os i. Vi fortæller med andre ord, at vi ønsker at starte spillet i Rum nr. 3. Hvis du ønsker at starte det i fx Rum nr. 8 – så værsgo. Så skal du bare skrive LO=8.

Prøv at taste den ovenfor anførte og analyserede program-sektion ind på din computer og kød (RUN) den. Skete der noget? Forhåbentlig ikke. Eftersom der ikke er PRINT-instruktioner i program-sektionen, skal computeren kun svare med et READY eller – på andre maskiner – OK.

Vi mangler altså nogle PRINT-instruktioner for at få computeren til at skrive vores DATA ud på skærmen. Dem får vi i næste sektion, som følger her. Men for at få denne sektion til at hænge sammen med den, vi lige har tastet ind, behøver vi en enkelt linie:

```
460 GOSUB 920 :REM GA NED TIL SKÆRMTEKST-SUBROUTINEN
*****
```

Tast den og derefter følgende linier ind:

```
919 REM SKÆRMTEKST-SUBROUTINE *****
920 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****
930 PRINT "DU ER I ";L$(LO)
940 FOR I=1 TO GL:IF G(I)<>LO THEN 980
950 PRINT:PRINT "DU KAN SE:"
960 FOR I=1 TO GL:IF G(I)=LO THEN PRINT G$(I)
970 NEXT I:GOTO 990
980 NEXT I
990 PRINT:PRINT "DU KAN GA:"
1000 FOR I=1 TO 4:IF L(LO,I)<>0 THEN PRINT V$(I)
1010 NEXT I
1020 RETURN
```

Vi tager igen linierne en for en:

919: REM-linie, som (hvis man ønsker det) senere kan slettes. Bemærk, at jeg har ladet den ende med et nital, hvilket gør det nemmere for mig at huske, at jeg ikke må »henvise« til den. Hvis jeg fx i linie 460 havde skrevet GOSUB 919, ville programmet stadig fungere, dersom jeg ikke slettede REM-linien 919 senere. Men hvis jeg slettede den, ville computeren (som *er dum*), ikke selv kunne finde ud af at gå videre til linie 920, som jo i realiteten er den linie, der indleder subrutinen. Så hvis jeg bliver enig med mig selv om, at jeg ALDRIG vil skrive en linie, der ender på et nital, efter en GOTO- eller GOSUB-instruktion, og ALTID lader REM-linier, der står som overskrift til en subrutine, ende på et nital, kan jeg være temmelig sikker på, at jeg aldrig ender i denne frustrerende situation. Blot et godt råd.

920: CLS-instruktionen vil ikke fungere på fx en Commodore-computer, men på visse andre. Den betyder »Clear Screen« (Ryd skærmen). Har du den ikke i din BASIC-version, så check din manual og find ud af, hvad du skal bruge i stedet.

930: Maskinen skriver DU ER I og derefter beskrivelsen af det rum, du befinder dig i, afhængigt af variabelen LO. Når vi begynder, er LO=3 (se linie 450), og vi befinder os i Rum nr. 3, og der vil på skærmens øverste linie komme til at stå DU ER I RUM NR. 3. Hver gang, vi fortæller maskinen, at vi flytter os (fx med inputtet GÅ SYDPA), ændrer den LO, og på skærmen vil vi komme til et nyt rum. Men mere om det i gennemgangen af tredje sektion.

940: En løkke, der beder computeren kigge efter i DATAsættet G(), om der befinder sig en genstand i det rum, vi er i. Hvis samtlige variabler i G() er forskellige fra LO, hopper vi ned til efter linie 980, og der bliver ikke skrevet noget på skærmen. Men hvis der *befinder* sig en eller flere genstande i rummet, fortsætter computeren til næste linie:

950: Her skriver computeren DU KAN SE efterfulgt af et kolon. Når der står PRINT to gange, betyder det blot, at vi laver en tom linie (et mellemrum) for overskuelighedens skyld.

960: Her skriver maskinen så, *hvad* du kan se (dvs. hvilke genstande, der befinder sig i det rum, du er i). Løkken gennemgår DATAsættet G(), og hvis nogen af variablerne har værdien LO, udskrives den tilsvarende variabel i strengsættet G\$() (som jo er specielt beregnet til skærmudskrifter). På skærmen kunne fx komme til at stå:

DU KAN SE:
ET SVÆRD
ET ÆBLE
ET SPEJL

970: Her kommer det NEXT, der afslutter løkken i linie 960, og vi får besked på at hoppe næste linie over, for den bruges kun, hvis der *ikke* er nogen genstande til stede i rummet (se linie 940).

980: Se forklaring til linie 940.

990: Endnu et ekstra PRINT for at lave et mellemrum. Computeren vil på skærmen skrive DU KAN GÅ plus et kolon.

1000: En løkke, der gennemlæser DATAsættet L() for at se, hvilke retninger man kan gå i. Hvis computeren støder på et nul, skriver den ikke noget ud på skærmen. Hvis den støder på et tal, anfører den det verdenshjørne, tallet henviser til med oplysninger hentet fra DATAsættet V\$, der er specielt beregnet til skærmudskrift.

1010: Det uundgåelige NEXT, der kræves hver gang man bruger en FOR-instruktion.

1020: Det uundgåelige RETURN (glem det aldrig!), der viser, at subrutinen er færdig, og i dette tilfælde sender os tilbage (RETURNer os) til linie 460.

Hvis du ikke allerede har gjort det, så tast denne program-sektion ind (naturligvis i samme program som det, du allerede har tastet ind – kald det hvad du vil, fx DEMO) og kørs (RUN) det hele. Så skulle du gerne få følgende udskrift på skærmen:

DU ER I RUM NR. 3

DU KAN SE:

ET SVÆRD

DU KAN GÅ:

SYDPA

ØSTPA

VESTPA

HVAD VIL DU GØRE?

Du skal endnu ikke fortælle computeren, hvad du vil gøre, for det vil bare resultere i en fejlmeddelelse. Men prøv endelig, hvis du ikke tror på mig!

Okay, hvordan skal vi så fortælle computeren, hvad vi vil gøre? Vi har allerede indlæst to DATAsæt, der gør den i stand til at identificere en række udsagnsord og navneord (se linie 430 og linie 440 og forklaringen til dem). Vores plan er nu at sørge for, at dersom vi skriver fx TAGE, skal computeren kunne genkende ordet (eller rettere de tre første bogstaver: TAG – for det er den eneste del af ordet, vi vil sørge for, at den behøver kende for at kunne identificere det). Og hvis vi skriver SVÆRD, skal den også kunne forstå det (de tre første bogstaver: SVÆ). Men hvad hvis vi skriver TAG SVÆRD – kan vi så få computeren til at forstå, at vi ikke bare har skrevet TAG – da den jo ser bort fra andet end de tre første bogstaver? Ja . . . for før vi lader den pille disse tre bogstaver ud, vil vi sætte den til at læse hele inputtet og fortælle den, at et tomt mellemrum betyder, at der kommer et nyt ord. Kan dette lade sig gøre? Det ville være rart, ikke? Tænk, hvis konversationen med maskinen skulle foregå på denne måde:

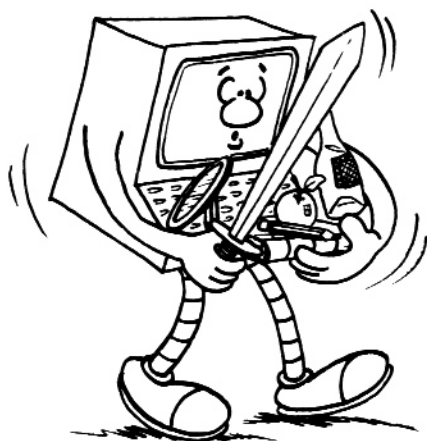
HVAD VIL DU GØRE?

Tage

TAGE HVAD?

Sværd

Det ville unægtelig være nemmere og mere elegant, hvis vi kunne få den til at forstå en enkelt sætning: TAGE SVÆRD. Og det kan vi. Faktisk vil vi kunne få den til at forstå meget længere sætninger – men jo længere de bliver, jo mere programmering (og hukommelse) skal der til. Så lad os nøjes med kommandoer på højst to ord – tro mig, det fungerer faktisk ganske tilfredsstillende.



For at få computeren til at forstå sætninger skal vi lave, hvad der kaldes en strengdelings-rutine. Vi skal 1) få den til at se, at der er mere end ét ord – 2) bede den kun læse de tre første bogstaver af disse ord (for at spare plads i vores DATAsæt) – men vi skal også 3) have den til at reagere på enkeltord (som fx vores STATUS) og 4) ord på mindre end tre bogstaver (som fx GÅ) – og vi skal 5) have den til at skelne mellem navneord og udsagnsord.

Det sidste lyder meget vanskeligt, men det er det faktisk overhovedet ikke. For i den eventyrspilmodel, vi opererer med, vil udsagnsordet altid komme først og navneordet altid komme sidst. TAGE SVÆRD, LÆGGE BOG, GÅ NORDPÅ – udsagnsord først, navneord sidst (vi husker, at vi af bekvemmelighedshensyn bilder computeren ind, at biordet NORDPÅ er et navneord).

Men lad os nu se på denne del af programmet og analysere det linie for linie:

```
470 PRINT:PRINT "HVAD VIL DU GØRE";  
480 INPUT B$  
490 PRINT  
499 REM STRENGDELINGS-RUTINE *****  
500 IF B$="R" THEN 460:REM RYD SKÆRM OG VIS  
    LOKALITETSBEKRIVELSE IGEN *****
```

```

510 IF B$="STATUS" THEN 860
520 IF B$="N" THEN B$="GA NORDPA"
530 IF B$="S" THEN B$="GA SYDPA"
540 IF B$="Ø" THEN B$="GA ØSTPA"
550 IF B$="V" THEN B$="GA VESTPA"
560 FOR I=1 TO LEN(B$)
570 IF MID$(B$,I,1)=" " THEN 600
580 NEXT I
590 GOTO 1030:REM HVIS KUN EET ORD *****
600 UO$=LEFT$(B$,I-1)
610 NO$=RIGHT$(B$, (LEN(B$)-I))
620 IF UO$="GA" THEN 670:REM IKKE HVIS A=AA *****
630 IF LEN(UO$)<3 THEN 1030
640 UO$=LEFT$(UO$,3):FOR I=1 TO AU:IF UO$=U$(I) THEN
680
650 NEXT I
660 UO=0:GOTO 1030
670 UO=1:GOTO 690:REM IKKE HVIS A=AA *****
680 UO=I
690 IF LEN(NO$)<3 THEN 1030
700 NO$=LEFT$(NO$,3):FOR I=1 TO AN:IF NO$=N$(I) THEN
730
710 NEXT I
720 NO=0:GOTO 1030
730 NO=I
739 REM UDSAGNSORD *****
740 ON UO GOTO 750,790,830

```

470: Vi laver for overskuelighedens skyld et mellemrum (det »tomme« PRINT) og lader computeren spørge os, hvad vi vil gøre.

480: Her kommer vores svar. Det kan fx være TAGE SPEJL eller LÆGGE FLASKE eller STATUS eller for den sags skyld GÅ AD HELVEDE TIL! eller DET VAR EN LØRDAG AFTEN. De to

sidstnævnte svar passer naturligvis ikke ind i vores model, men vi vil, som du længere nede kan se det, tage højde for »kvikke« bemærkninger.

490: Vi laver et mellemrum.

499: En REM-linie, der bagefter kan slettes.

500: Hvis du indtaster R (for Ryd skærmen), viser computeren os igen, hvor vi er. Dette kan være praktisk, når steds- og retningsangivelserne er forsvundet øverst på skærmen.

510: STATUS er kun ét ord og skal ikke deles. Så vi giver computeren besked på at gå ned til den subrutine, der viser os, hvad vi bærer på. Nu er vi parat til at tage mod kommandoer, der består af flere ord.

520-550: Når vi bevæger os rundt i et eventyrspils »landskab«, kan det være irriterende hele tiden at skulle skrive GÅ NORDPÅ, GÅ SYDPÅ etc. (især hvis spillet består af flere hundrede lokaliteter, som man gerne vil kunne bevæge sig hurtigt rundt imellem). Disse linier fortæller computeren, at dersom vi skriver N, betyder det GÅ NORDPÅ – dersom vi skriver S, betyder det GÅ SYDPÅ etc. Det sparer tid, især hvis man ikke er så hurtig til at indtaste.

560-580: Så er vi kommet til den egentlige strengdelingsrutine. I disse linier laver vi en løkke, der aflæser input B\$ i dets fulde længde (engelsk LENgth) og konstaterer, om der findes et mellemrum. Gør der det, hopper computeren ned til linie 600.

590: Her havner vi, hvis der ikke var noget mellemrum i input B\$. Så hopper vi ned til linie 1030. Lad os lige kigge på denne del af programmet:

```
1029 REM "OM IGEN"-ROUTINE *****  
1030 PRINT "OM IGEN!":GOTO 470
```

Linie 1029 er en REM-linie, der kan fjernes, hvis man ønsker det, og linie 1030 skriver OM IGEN! og hopper tilbage til linie 470: HVAD VIL DU GØRE? Grunden til, at jeg har lavet denne subrutine, er pladshensyn. Der spares hukommelse, når man i stedet for at skrive PRINT »OM IGEN!«:GOTO 470 kan nøjes med at skrive GOTO 1030, dog kun hvis man har brug for det mere end et par gange i programmet. Husk at tage linie 1029-1030 med, når du indtaster programmet.

600: Her fortæller vi maskinen, at UO\$ (udsagnsordet) er alt til venstre for mellemrummet (I'et fra løkken i linie 560-580).

610: Og her fortæller vi maskinen, at NO\$ (navneordet) er alt til højre for mellemrummet.

620-630: Vi har endnu ikke bedt vores computer pille de tre første bogstaver ud af henholdsvis udsagnsordet og navneordet, og inden vi gør det, skal vi rydde ord på mindre end tre bogstaver af vejen, så der ikke går koks i maskineriet. Vi opererer kun med ét (GÅ), og dersom det dukker op (og det kommer det til at gøre nogle gange i løbet af spillet), *har* vi identificeret vores udsagnsord og kan springe ned til linie 670 (se denne). Dersom du ikke har bolle-å på din maskine, men er nødt til at skrive GAA med to a'er, hvorved ordet jo bliver på tre bogstaver, skal du bare sløjfe linie 620. I linie 630 skiller vi os af med alle andre udsagnsord på under tre bogstaver ved at henvise til linie 1030 (OM IGEN-rutinen).

640-650: Her gør vi det, at vi korter UO\$ (udsagnsordet) ned til de tre bogstaver, maskinen skal operere med for at kunne genkende det i DATAsættet U\$. Hvis det bliver genkendt, hopper vi ned til linie 680 (se denne).

660: Hvis computeren ikke har fundet udsagnsordet i DATAsættet U\$, betyder det, at den ikke kender det. Hvis vi for eksempel havde skrevet DET VAR EN LØRDAG AFTEN, vil maskinen på dette punkt have isoleret DET som udsagnsordet (også selv om det ikke *er* et udsagnsord). Den genkender ikke ordet, og det får

(i skikkelse af talvariablen UO) værdien 0 og bliver forkastet. Vi går ned til OM IGEN-rutinen.

670: Her havner vi fra linie 590, hvor vi har identificeret udsagnsordet som GÅ. Vi giver dets talbroder UO (som er nemmere for computeren at arbejde med end en strengvariabel) værdien 1. Det betyder, at GÅ står først i vores række af udsagnsord – og det vidste vi jo godt i forvejen. Vi hopper videre til 690, for nu kommer den oplysning, der står i linie 680, ikke os ved.

680: Her tilskrives variablen UO den værdi, der er blevet fundet frem til via løkken i linie 640-650. UO svarer stadig ganske til UO\$, men er som nævnt en talkode for den, som vi er nødt til at indføre, for at computeren kan arbejde effektivt med den. Hvis UO\$ fx (via forkortelsen i linie 640) er LÆG, bliver UO=3 (hvilket svarer til ordet LÆGs plads i DATAsættet U\$()). Lad os for at gøre det helt overskueligt lige slå fast, at

U\$() = Det DATAsæt, der indeholder udsagnsord, computeren skal kunne genkende via inputtet B\$.

UO\$ = Den strengvariabel, der (i sidste ende) indeholder de tre første bogstaver i udsagnsordet (første ord) i input B\$, dersom dette består af mere end et ord.

UO = Den talvariabel, som computeren når frem til ved at sammenligne UO\$ med U\$, og som den skal bruge, når den skal videre i programmet (se linie 740).

690-730: Her behandler vi navneordet (det andet ord i inputtet B\$) ganske som vi behandlede udsagnsordet. Det ender med, at vi genkender navneordet og giver det værdien NO. Hvis vi ikke genkender det, er NO=0, og så hopper vi ned til OM IGEN-rutinen (hvis vi fx havde skrevet GÅ AD HELVEDE TIL, ville maskinen – der kun reagerer på det *første* mellemrum – have identificeret »navneordet« som AD HELVEDE TIL og forkastet det).

739: REM-linie, der kan smides væk efter behag.

740: Her får maskinen at vide, hvor den skal gå hen, alt efter hvilket udsagnsord, vi har givet den. Har vi sagt GÅ, er UO=1, og maskinen går til linie 750 (GÅ-rutinen). Har vi sagt TAGE, er UO=2, og maskinen hopper ned til linie 790 (TAGE-rutinen). Og har vi sagt LÆGGE, er UO=3, og vi fortsætter fra linie 830 (LÆGGE-rutinen).

Lad os rekapitulere, hvad vi indtil nu har gjort: Vi har fodret computeren med DATA, vi har givet den anvisninger på, hvordan skærbilledet skal se ud, og vi har lært den at forstå vores input.

Nu mangler vi blot at få den til at *reagere* på vores input – at skabe en illusion af, at vi bevæger os rundt i vores landskab, at vi tager og lægger de ting, vi støder på. Vi mangler at give den de sidste ordrer, der gør det hele levende.

Vi ved, at prøvespillet opererer med fire grundkommandoer:

1. GÅ + Retning
2. TAGE + Genstand
3. LÆGGE + Genstand
4. STATUS

I den programsektion, vi lige gennemgik, har vi allerede henvist computeren til de linier, der hører til hver af disse grundkommandoer. Så lad os nu kigge på disse linier (de sidste i demonstrationsprogrammet). Vi tager én grundkommando ad gangen:

```
749 REM GA *****
750 IF NO<7 THEN 1030
760 NO=NO-6
770 IF L(LO,NO)=0 THEN PRINT "DU KAN IKKE GA DEN
    VEJ.":GOTO 470
780 LO=L(LO,NO):GOTO 460
```

749: REM-linie, der kan slettes efter behag.

750: Hvis navneordet (NO) er mindre end 7 (dvs. hvis det ikke er et verdenshjørne), går vi ned til OM IGEN-subrutinen. Hvis du fx skriver GÅ BLYANT, får du besked på at prøve noget andet.

760: Her omnummererer vi vores tilbageblevne »navneord«, der har værdierne 7-10, og giver dem værdierne 1-4. Det gør vi for i de næste to linier at kunne sammenligne dem med DATAsættet L(), der netop opererer med fire verdenshjørner nummereret fra 1-4. Hvis du fx har skrevet GÅ ØSTPÅ, vil ØSTPÅ, der før havde talværdien 9, nu have talværdien 3, som computeren kan bruge i de instruktioner, vi giver den i de næste to linier. I øvrigt er denne linie en af dem, du ikke bare kan skrive af, hvis du begynder at lave om på antallet af navneord. Hvis du fx har lavet et spil, der opererer med 20 navneord, nytter det ikke noget, at du så skriver $NO=NO-6$, da denne formel kun gælder for et sæt på 10 navneord. Du burde skrive $NO=NO-16$. Husk det! Dette er et af de steder, der virkelig kan gå kuk i den, når du skal lave dit eget program. Husk, at det er antallet af *de navneord, der ikke er verdenshjørner*, du skal trække fra her (hvis du vel at mærke har anbragt verdenshjørnerne til sidst. Du kunne også anbringe dem *først*, hvilket ville være det ideelle. Så behøver du slet ikke denne linie. Når jeg har anbragt dem til sidst, er det for at anskueliggøre problemet, der kan volde store bryderier, hvis man ikke er opmærksom på det).

770: Her giver vi vores maskine ordre til at sammenligne vores ønske med DATAsættet L(). Hvis den konstaterer, at verdenshjørneangivelsen NO på lokaliteten LO er lig nul, fortæller den dig: DU KAN IKKE GÅ DEN VEJ og spørger dig endnu en gang: HVAD VIL DU GØRE? (Linie 470).

780: Variablen LO får værdien, som står i L(LO,NO), hvis der altså ikke står et nul der. Det vil sige, at når vi hopper tilbage til linie 460 og derfra ned til skærmttekst-subrutinen, har vi et nyt LO (det vil sige en ny Lokalitet), og den tekst, der bliver skrevet ud, vil have »flyttet« spilleren til et nyt sted.

Så nemt var det altså at GÅ. Hvis du ikke allerede har gjort det, så tast GÅ-sektionen ind sammen med det, du indtil videre har skrevet af, og se, om den virker i praksis. Hvis den ikke gør det, har du begået en indtastningsfejl!

Vi går videre til næste grundkommando (TAGE):

```
789 REM TAGE *****
790 IF NO>6 THEN 1030
800 IF G(NO)=-1 THEN PRINT "DET HAR DU ALLEREDE
      GJORT.":GOTO 470
810 IF G(NO)<>LO THEN PRINT "FINDES IKKE HER":GOTO
      470
820 G(NO)=-1:PRINT "UDMÆRKET.":FOR I=1 TO
      500:NEXT I:GOTO 460
```

790: Her smider vi de »navneord«, der ikke er genstande (dvs. verdenshjørnerne), væk. Vær også her opmærksom på, at 6-tallet kun passer på et DATAsæt på 10, hvor verdenshjørnerne er opført til sidst. Hvis du laver dit eget spil med et andet antal navneord, så husk, at dette tal skal laves om.

800: Vi husker, at DATAsættet G() er det, der holder rede på, hvor tingene befinder sig. Hvis vi fx skriver G(2)=3 betyder det, at Genstand nr. 2 ligger i Rum Nr. 3. Men her beder vi maskinen checke, om genstanden har værdien minus 1. Hvad skal det betyde? Vi har da ikke noget rum, der har værdien minus 1, vel?

Nej, dette »rum« er *dig*. Som du kan se i Linie 820, fortæller vi computeren, at dersom du TAGer en ting, skal den overflyttes til Rum nr. -1. Så ved den, at du *har den på dig*. Tallet -1 er tilfældigt valgt – det kunne for dens sags skyld lige så godt være 2000, men for ikke at forveksle det med numrene på de andre rum giver vi det en negativ værdi.

Computeren får altså at vide, at dersom du har den ting, du ønsker at TAGE, på dig, så skal den skrive: DET HAR DU ALLE-

REDE GJORT og derefter igen spørge dig: HVAD VIL DU GØRE?

810: Her fortæller computeren dig, at den ting, du vil tage, *ikke* befinder sig på stedet. Hvis G(3) fx er 5 (hvis spejlet er i Rum nr. 5), og *du* er i Rum nr. 6, så får du beskeden: FINDES IKKE HER.

820: Hvis computeren overhovedet når til denne linie, ved den, at du ikke har genstanden på dig, og at den ikke befinder sig et andet sted – at du har lov til at tage den. Genstanden får værdien -1 (hvilket betyder, at du har den på dig – se ovenstående linie), og der kvitteres med et UDMÆRKET, som får lov til at stå et øjeblik (takket være en »tom« løkke), før skærmen ryddes. Dette UDMÆRKET fortæller dig, at din ordre er godtaget, og at du nu har genstanden på dig.

Lad os fluks se på, hvordan vi LÆGger tingene fra os igen:

```
829 REM LÆGGE *****
830 IF NO>6 THEN 1030
840 IF G(NO)=-1 THEN PRINT "UDMÆRKET.":FOR I=1 TO
      500:NEXT I:G(NO)=LO:GOTO 460
850 PRINT "FOR AT KUNNE LÆGGE NOGET":PRINT "MÅ MAN
      HAVE DET PÅ SIG.":GOTO 470
```

830: Her smider vi igen verdenshjørnerne væk (se forklaring til Linie 790).

840: Hvis du har genstanden på dig (hvis G(NO)=-1), så bliver den lagt – G(NO) får værdien LO, og genstanden befinder sig på stedet. Der kvitteres med et UDMÆRKET, der (via en »tom« løkke) får lov til at stå et øjeblik. .

850: Computeren meddeler dig surt, at FOR AT KUNNE LÆGGE NOGET MÅ MAN HAVE DET PÅ SIG. Den ved nemlig, at

dersom den overhovedet kommer til denne linie, så er det fordi du ikke er blevet ekspederet tilbage til linie 460 i den forrige linie – den ved, at du ikke har den anførte genstand på dig og har begået en fejl ved at meddele, at du vil LÆGge den.

Og hvordan ser vi så, *hvad vi har på os*? Det gør vi ved hjælp af kommandoen STATUS, som bliver effektueret her:

```
859 REM STATUS *****
860 PRINT:PRINT "DU BÆRER FØLGENDE: "
870 BP=0
880 FOR I=1 TO GL:IF G(I)=-1 THEN PRINT G$(I):BP=1
890 NEXT I
900 IF BP=0 THEN PRINT "INGENTING"
910 GOTO 470
```

860: Computeren laver et mellemrum (tom linie) og skriver DU BÆRER FØLGENDE efterfulgt af et kolon.

870: Af bekvemmelighedshensyn indfører vi en ny variabel som vi kalder BP (Bærer På). BP gør det lettere for os at udføre de næste linier. Vi nulstiller den, så computeren ikke bruger den værdi, den måske fik, sidste gang vi kom forbi her.

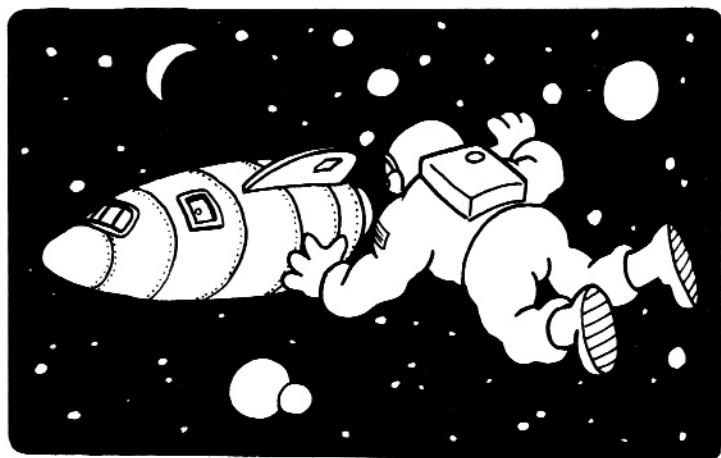
880-890: En løkke, der beder computeren lave en liste over de ting, der har værdien -1 (altså dem, vi har på os). Vi sætter her BP til 1, for så bliver Linie 900 (se denne) sprunget over.

900: Hvis vi ikke har noget på os, har BP ikke fået værdien 1 i forrige linie og er stadig 0 – og computeren fortæller os, at vi bærer på INGENTING.

910: Vi går tilbage til Linie 470 (HVAD VIL DU GØRE?)

Og hermed er vores prøveprogram komplet. Kør det igennem, indtil du er fortrolig med mekanismerne i det. Hvis du får fejlmeddelelser, har du – måske – begået en eller flere indtastningsfejl. Ret dem!

3 · Det mystiske rumskib



En af de ting, jeg tangerede i begyndelsen af forrige kapitel, var handlingen . . . for det nytter ikke så meget, at vi kan gå rundt i et landskab med ti eller måske hundrede lokaliteter og samle ting op, hvis vi ikke har nogen idé – noget mål – med det. De fleste af de eventyrspil, der findes på det engelske og amerikanske marked, foregår i mystiske sagnriger (à la »Ringenes herre«), hvor hovedpersonen som regel skal finde en eller anden skat og undervejs slås med diverse forhindringer. Der er imidlertid ikke det fjerneste i vejen for at lave en spændende historie, der fx foregår i København og handler om at nå ind til Hovedbanegården, inden toget til Århus kører. Man kan også lave eventyr, der faktisk er kriminalgæder, hvor spilleren er detektiven og skal opklare et mord (Se »Mordet i den lukkede boks« sidst i denne bog). Eller et, hvor spilleren skal nå at finde og demontere en atombombe, som nogle terrorister har anbragt et sted i en by . . . Og hvis man synes, at al denne virkelighedsflugt er af det

onde, kan man lave et »eventyr«, der går ud på at udrydde Danmarks betalingsbalanceunderskud i løbet af 5 år. Kun fantasien sætter grænserne for, hvilken handling man vil give sit eventyr, og på dette punkt kan jeg ikke hjælpe dig . . . du må bruge din egen fantasi.

Men jeg kan forsøge at afstikke nogle retningslinier for, hvordan man laver en handling. Jeg har allerede nævnt, at det øger spændingen at have en del lokaliteter, som man kan udforske – det er i sig selv interessant, men naturligvis ikke nok. Der skal også ind imellem *ske* noget, som skaber en eller anden *forandring*, så spillet *udvikler* sig. I vores demonstrationsspil så vi, hvordan vi ved at ændre variabelernes værdi kunne skabe en illusion af, at vi bevægede os fra sted til sted eller tog eller lagde forskellige ting. Lad os nu kigge på, hvordan vi (også ved hjælp af variable) kan sætte lidt skub i en handling.

Først må man have en idé til en handling. Den bør være hundrede procent udbygget, før man går i gang – om end erfaringen viser, at man ofte finder på sjove ting undervejs, så lav altid *god* plads mellem linierne!

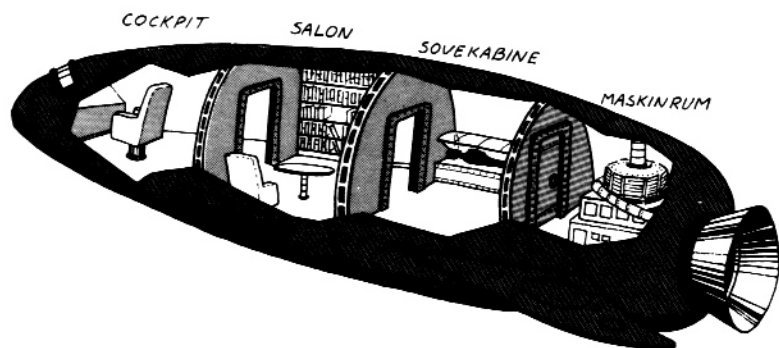
Allerførst må man ifølge sagens natur sætte sig et eller andet mål. Dette mål kan som sagt være alt muligt lige fra at nå ind på Hovedbanegården og til at finde ud af, hvem der anbragte kniven i ryggen på liget i biblioteket. Når man har fundet sit mål, konstruerer man en handling ud fra det, og når man har sin handling parat, kan man faktisk ved hjælp af forholdsvis simple hjælpemidler få den til at fungere.

Jeg tror, jeg bedst kan illustrere det for dig ved hjælp af et nyt (men langt mere udbygget) demonstrationsspil, som faktisk *er* et færdigt eventyrsspil. Da jeg allerede *har* vist dig, hvordan man laver lokaliteter, vil jeg koge denne del af det nye spil ned til et minimum og koncentrere mig om de ting, du endnu ikke har lært. Så jeg vil nøjes med fire lokaliteter – så kan du også se, at det sagtens kan lade sig gøre at lave et sjovt spil uden at behøve spille dyrebar hukommelsesplads på alenlange lokalitetsbeskrivelser.

Da jeg lavede spillet, bestemte jeg, at hovedhandlingen skulle være, at spilleren er strandet med sit rumskib ud for en planet og skal finde ud af, hvordan han får det i gang igen, så han kan

lande sikkert. Jeg besluttede, at rumskibet (af de ovenfor nævnte grunde) kun skulle have fire »lokaliteter«.

Lad os kigge på et kort over disse fire rum:



Allerede på dette punkt blev jeg klar over, at det ikke var særlig spændende at udforske sit *eget* rumskib. Så jeg besluttede mig til at indlede historien med, at spilleren er strandet ud for en planet i sit rumskib, at dette ikke kan repareres, og at han er nødt til at svæve over til et *andet* rumskib, der ligesom hans eget ligger i kredsløb om planeten. Vores kort forestiller altså et *fremmed* rumskib, og spillet begynder, da vor helt er kommet ind gennem luftslusen i dette.

For at gøre det svært for ham allerede fra begyndelsen fortæller vi ham, at ilten i hans rumdragt er sluppet op, og at han ikke kan forlade det fremmede rumskib, før han er landet med det.

Hermed har vi uigenkaldeligt forhindret spillet i at brede sig til flere end de fire lokaliteter, vi allerede har bestemt os for.

Og nu er det, vi skal til at bruge fantasien for alvor. For hvad kan forhindre vor helt i at gå ind i cockpittet, starte motoren og flyve ned til planeten og afslutte spillet, inden det overhovedet er begyndt?

Jo . . . vi kunne fx meddele ham, at motoren er gået i stykker, og at han er nødt til at reparere den. Hermed har vi allerede skabt et problem, han skal løse. Lad os sige, at der er røget en sikring i maskinrummet, som han er nødt til at skifte ud. Vi anbringer en frisk sikring i salonen, som han kan tage.

Og så går han ud i maskinrummet og skifter sikringen ud og starter rumskibet, ikke?

Nej! For så ville spillet være forbi. Vi må have nogle flere forhindringer. Vi begynder med at låse døren ud til maskinrummet, så han er nødt til at finde en måde at åbne den på, før han kan gå derud.

Måske har du allerede spekuleret på, hvad et *tomt* rumskib laver i kredsløb om planeten. Det er ikke videre logisk, vel? Lad os give det en besætning. For en nemheds skyld gør vi det til et enmandsrumskib, så der kun er ét besætningsmedlem. Lad os kalde ham »kaptajnen«.

Men hvorfor har kaptajnen ikke selv repareret sin motor? Hmm . . . lad os sige, at han ikke *ved*, den er gået i stykker. Han sover – eller ligger måske i dvale – inde i sovekabinen.

Men hvad kan forhindre vor helt i at vække kaptajnen og bede ham om hjælp? Vi kunne finde på mange forklaringer – han kunne være syg, han kunne være død, han kunne være fjendtligt indstillet.

Vi vælger den sidste mulighed, for derved kan vi introducere endnu et spændingsmoment: en fjende. Vi vedtager, at dersom vor helt vækker kaptajnen, bliver han overfaldet af ham og slået ihjel.

Men så kan han jo bare lade være med at vække kaptajnen, ikke?

Det er der også råd for. Vi anbringer »nøglen« til døren (i form af et magnetkort) i lommen på kaptajnen, så vor helt er *nødt* til at vække ham for at få fat i den.

Nu har vi for alvor fået gang i handlingen – i spillet: vi har givet vores helt indtil flere problemer, der skal løses. Men det skulle jo gerne være rigtig spændende og indviklet, så lad os give ham nogle flere, for indtil videre kan han jo bare slå kaptajnen ihjel uden at vække ham og tage magnetkortet, åbne døren, reparere motoren og starte rumskibet. Lad os prøve at gøre det svært for ham at slå kaptajnen ihjel. Lad os give kaptajnen overmenneskelige kræfter. Lad os køre ud på fantasiens overdrev og gøre ham til *vampyr*!

Som du ved, findes der visse afstukne regler for bekæmpelse af vampyrer: man skal have en hammer, og man skal have en

træpind, man kan hamre i hjertet på ham. Så lad os anbringe en hammer et sted på rumskibet og noget af træ, han kan slå i stykker, så han får et træstykke, han kan bruge til at dræbe vampyren med. Lad os også være fair over for ham og fortælle ham, hvordan man slår en vampyr ihjel, hvis han ikke skulle vide det i forvejen. Det kan vi meget elegant gøre ved i salonen at anbringe en reol af træ, i hvilken der bl.a. står et leksikon, hvori han kan læse om vampyrer.

Så vidt, så godt. Hvis vor helt finder ud af at slå en flis af reolen, kan han altså dræbe vampyren, tage magnetkortet, åbne døren, reparere motoren og starte rumskibet.

Det er for nemt, synes du ikke?

Okay . . . Så lad os understrege, at vampyren har overmenneskelige kræfter, og at han ikke kan overmandes, hvis man ikke finder ud af at *svække* ham, før man bruger hammeren og træstykket.

Og da vist nok alle er klar over, at vampyrer ikke kan døje kors, ja, faktisk får det rigtig dårligt, hvis man stikker sådan et op i hovedet på dem, anbringer vi et sted på rumskibet et krucifiks, som man kan svække ham med.

Men er det ikke stadig for nemt? Vor helt kan jo bare tage krucifikset, svække vampyren, dræbe ham med hammeren og træstykket og så videre og så videre. Det er stadig væk lidt for forudsigeligt, ikke? Hvordan kan vi gøre det mere surt for vores stakkels helt?

Jo, vi kan vedtage, at krucifikset *ikke virker*. Hvis vor helt prøver at svække vampyren med krucifikset, får han sig en meget ubehagelig overraskelse: han bliver slået ihjel.

Men hov! Nu bryder vi alle vedtagne regler for vampyrbekæmpelse, ikke? Enhver ved da, at vampyrer ikke kan *fordrage* kors!

Jo . . . men grunden til, at de ikke kan fordrage kors må være, at korset er et godt gammeldags kristent symbol, og at vampyren er en af Satans skabninger, der ikke er på allerbedste fod med Vorherre og i rædsel viger tilbage for »fjenden«. Men hvad nu, hvis vampyren ikke er kristen? Så har han ikke noget forhold til et kors og kan ikke blive bange for det. Vi kunne gøre ham til jøde – så burde han være bange for en davidsstjerne, der er jø-

dernes hellige symbol . . . men ville vi få så forfærdelig meget ud af at anbringe en davidsstjerne på rumskibet? Ville vor helt ikke hurtigt finde ud af at bruge den i stedet for korset?

Jo, givetvis . . . vi bliver nødt til at være endnu mere snedige. Vi gør vampyren til muhamedaner – eller i hvert fald *frafalden* muhamedaner. Muhamedanernes hellige symbol er halvmånen . . . så vi vedtager, at dersom vi kan true ham med en *halvmåne*, kan vi svække ham.

Så vi anbringer altså en halvmåne et sted på rumskibet . . . eller hvad? For gør vi det, er vi lige vidt. Den skal vor helt nok temmelig hurtigt finde ud af at bruge.

Så hvad gør vi?

Nu er det så heldigt, at vi befinder os ude i rummet tæt på en planet, og planeter har som bekendt tit og ofte måner, så vi anbringer i al vores durkdrevenhed en måne uden for rumskibet, og ingen forhindrer os i at gøre den halv. Vi vedtager, at der kun er ét vindue i rumskibet (i cockpittet), og at man er nødt til at lokke vampyren hen til vinduet for at svække ham. Og vi benytter lejligheden til at gøre det endnu sværere for vor helt ved at lukke vinduet med skodder, der skal åbnes, før man kan kigge ud – og vi drejer rumskibet væk fra månen, så vor helt selv er nødt til at dreje det rundt for at få månen frem. Selv om vi allerede har vedtaget, at motoren er gået i stykker, kan vi godt anbringe nogle små styreraketter på siden af rumskibet, hvis de ikke kan gøre andet end at *dreje* det – så lad os gøre det.

Nu kan vor helt altså slå skodderne fra vinduet og dreje rumskibet om sin egen akse (men *ikke* starte hovedmotoren) og få en halvmåne frem på himlen uden for vinduet – en halvmåne, der vil svække vampyren tilstrækkeligt til, at man kan slå ham ihjel med hammeren og træstykket, *hvis han vel at mærke befinder sig i cockpittet*. Dersom vor helt prøver at slå ham ihjel alle andre steder, vil han selv lide døden, og dersom han ikke har fået månen frem uden for vinduet, vil han *også* dø, selv om det skulle lykkes ham at lokke vampyren ind i cockpittet.

Og hvordan får vi så lokket vampyren derind? Ved at råbe »Komsedasse, lille pyr«? Det var naturligtvis en udvej, men selv om vi befinder os på fantasiens overdrev, kan vi jo godt prøve at bevare en *vis* indre logik, og enhver ved (eller burde vide), at

vampyrer ikke kan stå for lugten af blod, så lad os vedtage, at han skal lokkes ud i cockpittet med noget *blod*.

Og hvor finder man så blod om bord på et rumskib? Tja, vi kunne jo anbringe nogle flasker med det i et skab (det *er* jo trods alt vampyrens rumskib, og han *skal* jo have noget at leve af), men nej . . . det ville være alt, alt for nemt for vores helt at finde ud af at bruge det. Næ, lad os endnu en gang gøre det en smule svært for ham. Vi anbringer en kniv, så han kan tappe noget *frisk* blod . . . og vi benytter lejligheden til at lave endnu en fælde: hvis han prøver at skære i sig selv, lader vi ham forbløde og dø. Og når vi alligevel har fat i en kniv, kan vi også vedtage, at hvis han ikke har *spidset* træstykket med kniven, før han prøver at dræbe vampyren med det, vil han selv blive dræbt.

Men hvorfra får vi så *blodet*? Jo, vi laver endnu et lille intellektuelt problem, vores helt kan tumle med. Vi anbringer en *musefælde* i salonen og håber så, at han kan finde ud af at aktivere den, så der går en *rotte* i den – og det er denne rotte, vor helt skal dræbe for at skabe den liflige blodduft, der får vampyren til at komme farende.



Vi kunne sagtens gøre det endnu mere surt for vores hovedperson ved at anbringe flere fælder og ting om bord på rumskibet, men lad os stoppe her. Dette er jo kun et udvidet demonstrationsspil, og jeg er sikker på, at du på indeværende tidspunkt har fået en idé om, hvordan man bygger nogle nederdrægtige

problemer op, der tilsammen giver, hvad vi faktisk godt kan kalde en *handling*. Men lad os lige til sidst prøve at se, om vi ikke kan lave endnu mere ravage med de ting, vi *har*.

Jo, det kan vi skam godt. Vi ved allerede, at hvis vor helt ikke bruger tingene i en bestemt rækkefølge og på bestemte steder, vil det gå ham ilde. Lad os føje endnu et par ulykker til listen: hammeren, som vi ved skal bruges til at slå en flis af reolen og banke træstykket i hjertet på vampyren, kan også bruges til at smadre henholdsvis vinduet i cockpittet, luftslusen i salonen og motoren i maskinrummet, hvis man ikke tager sig i agt, og ilten vil strømme ud af rumskibet, hvis man forsøger at *aktivere* luftslusen – med eller uden hammer.

Godt . . . men lad os også være *fair* over for vores helt . . . vores spiller. Lad os endelig ikke holde noget skjult for ham – det ville være snyd og ret hurtigt fratage ham enhver lyst til at spille spillet færdigt. Husk, at spilleren *altid* skal have en rimelig mulighed for at løse problemerne!

Vi vil hjælpe ham lidt ved at anbringe en koran i bogreolen, så han kan regne ud, at vampyren er muhamedaner, og i koranen vil vi skrive, at halvmånen er muhamedanernes hellige symbol – ligesom vi i indledningen til spillet vil gøre ham opmærksom på, at der *er* en måne uden for rumskibet. Vi vil fortælle ham, at den mand, han finder i dvalekisten, er vampyr, og vi vil i det hele taget give ham mulighed for at *undersøge* hver enkelt ting om bord på skibet – selv om han ikke altid bliver klogere af det!

Så vidt, så godt . . . men hvordan får vi nu dette til at fungere i praksis? Tja, vi ved, at vi skal give vores computer nogle DATA for at få den til at forstå os. Lad os tage de DATA, vi skal bruge i dette spil, fra en ende af.

A. Retningsangivelser

Her kan vi nøjes med »forud« og »agterud« (vi befinder os på et skib, og rummene hænger sammen i en fortløbende kæde, der består af fire led).

B. Genstande

1. En måne: Selv om vi ikke kan TAGE og LÆGGE den, har vi brug for at UNDERSØGE den. Derfor tager vi den med i vores DATAsæt.
2. Et vindue: Skodderne skal kunne AKTIVERES, og vi skal kunne UNDERSØGE det for at se, om månen er udenfor.
3. En luftsluse: Har vi med for at kunne UNDERSØGE og AKTIVERE den (det sidste vil jeg kraftigt fraråde!).
4. En reol: Skal kunne UNDERSØGES.
5. En koran: Skal kunne TAGES, LÆGGES, UNDERSØGES og LÆSES.
6. Et leksikon: Skal opfylde samme funktioner som koranen.
7. En musefælde: Skal kunne AKTIVERES, TAGES, LÆGGES og UNDERSØGES.
8. En rotte: Skal kunne TAGES, LÆGGES og UNDERSØGES.
9. Et træstykke: Skal kunne TAGES, LÆGGES og UNDERSØGES.
10. En hammer: Skal kunne TAGES, LÆGGES, UNDERSØGES og BRUGES.
11. En kniv: Som hammeren.
12. En kiste: Det er den, vampyren ligger i, og den skal kunne UNDERSØGES.
13. En vampyr: Skal kunne UNDERSØGES.
14. Et magnetkort: Skal kunne TAGES, LÆGGES, UNDERSØGES og BRUGES.

15. En dør: Det er den, der er låst (mellem sovekabinen og maskinrummet), og den skal kunne UNDERSØGES.

16. En sikring: Skal kunne TAGES, LÆGGES, UNDERSØGES og BRUGES.

17. Et krucifiks: Skal kunne UNDERSØGES.

18. Et rumskib: Er taget med, fordi de fleste uvægerligt vil forsøge at AKTIVERE RUMSKIB eller UNDERSØGE RUMSKIB.

19. Et instrumentbræt: Skal kunne AKTIVERES og UNDERSØGES.

20. En mand: Skal kunne UNDERSØGES, så vi kan få at vide, at han er vampyr.

C. Udsagnsord

Du har allerede set de fleste af dem under *Genstande*, men her er den fulde liste:

1. GÅ
2. AKTivere
3. UNDersøge
4. BRUGe
5. LÆSe
6. LÆGGe
7. GEMme
8. FORtsætte
9. TAGe

Du vil sikkert undre dig over, hvorfor jeg ikke har medtaget fx DRÆbe (når man skal slå vampyren eller rotten ihjel), men vi sparer en masse hukommelsesplads ved at bruge så få udsagnsord som muligt, og disse ting kan lige så godt gøres ved fx at BRUGE HAMMER eller BRUGE KNIV.

GÅ siger sig selv (man skal kunne gå agterud og forud), hvorimod GEMme og FORTsætte kræver en nærmere forklaring: disse to kommandoer er udelukkende beregnet på at bruges sammen med ordet SPIL (som vi altså kan føje til vores liste af navneord). Som jeg vil vise dig, vil du på ethvert sted ved hjælp af disse kommandoer kunne GEMME SPILLET og senere FORTSÆTTE SPILLET fra det sted, du kom til. Det sker, ved at man opretter en fil, hvori variabelværdierne (på det tidspunkt, man gemmer dem) oplagres for siden at kunne hentes frem igen. Det har nok ikke den store betydning i dette lille spil, hvor man på få minutter kan gentage de manøvrer, man har foretaget, fra begyndelsen, men forestil dig, at du er i gang med et spil med 100 lokaliteter og har brugt en hel dag på at nå frem til fx Lokalitet nr. 43 og undervejs har samlet en masse ting op, som du skal bruge – hvorefter din familie så fortæller dig, at de skal se TV-avis, og at du værsgo har at slukke for den møgcomputer . . . ville det så ikke være rart at kunne fortsætte fra det sted, man kom til, i stedet for at være nødt til at begynde helt forfra?

Visse eventyrspil kan det tage flere måneder at nå igennem, så et raffinement som en gemme/fortsætte-rutine er yderst rar at have med. Jeg skal komme nærmere ind på, hvordan man bærer sig ad med at lave sådan en, i gennemgangen af selve spillet.

Men inden vi kommer så vidt, skal jeg da lige fortælle dig, hvordan vi bærer os ad med at få computeren til at forstå, at rotten er død eller træstykket er blevet spidset eller motoren er smadret. Det lyder, som om den skal holde rede på en pokkers masse ting, men i virkeligheden kan vi i dette spil ud over de DATAsæt, vi opretter, nøjes med at operere med ganske få variabler for (som det hedder på nudansk) at få det til at fise ind i hukommelsesregistrene. Mange af de ting, computeren har brug for at vide, kan vi fortælle den via de indekserede variabler, vi allerede har puttet ind i vores DATAsæt – hvis sikringen fx befinder sig i maskinrummet, ved computeren, at motoren er repareret, og så videre og så videre, men ud over dette er der visse ting, vi har brug for at fortælle den særskilt, så den kan træffe de rigtige valg. Det er ikke mange ting – sikkert færre end du umid-

delbart tror. Du er allerede fra det første demonstrationsspil fortrolig med den ikke-indekserede variabel LO, der fortalte computeren, i hvilket rum, vi befandt os. Den har vi også brug for i det spil, vi her beskæftiger os med, og derudover medtager vi følgende:

SK: har SK værdien 0, er skodderne for vinduet lukkede. Er værdien 1, er de åbne.

MU: Er værdien 0, er musefælden ikke aktiveret. Er værdien 1, er den aktiveret.

TR: Er værdien 0, er træstykket ikke spidset. Er værdien 1, er det spidset. Er værdien 2, er motoren smadret (vi kan ikke komme ud i maskinrummet og ved et uheld smadre motoren, før træstykket *er* spidset og vampyren dræbt, så hvorfor ikke bruge variabelen TR, som vi alligevel ikke har brug for mere, til at markere dette, i stedet for at oprette en ny variabel?)

DO: Markerer, om døren er åben eller lukket.

DR: Hvis værdien er 1, er rumskibet blevet drejet 30 grader, hvis den er 2, er rumskibet blevet drejet 60 grader etc. Der bliver lagt 30 grader til hver gang op til 360 grader, hvorefter variabelen atter nulstilles (for da er man nået hele cirklen rundt).

RO: Markerer, om rotten er levende eller død.

VA: Markerer, om vampyren er levende eller død.

Det er alt! Mere skal der ikke til for at få handlingen til at fungere (ud over naturligvis en hel del instruktioner, der fortæller computeren *hvordan* og *hvornår* den skal danse efter vores pibe). Her følger vores LISTige spil sektion for sektion med en gennemgang af de enkelte linier ind imellem:

```

9 REM DET MYSTISKE RUMSKIB *****
10 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
   CHR$(147) *****
20 PRINT "*****"
30 PRINT "      DET MYSTISKE RUMSKIB"
40 PRINT:PRINT "et eventyrspil for rumrejsende"
50 PRINT "*****"
60 FOR I=1 TO 8:PRINT:NEXT I
70 PRINT "NAR SPILLET ER GAET I GANG,"
80 PRINT "KAN DU TIL ENHVER TID"
90 PRINT "SKRIVE 'HJÆLP', HVIS DU"
100 PRINT "SKULLE VÆRE I TVIVL OM NOGET."
110 PRINT:PRINT "GOD FORNØJELSE!"
120 GOSUB 4120
125 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
   CHR$(147) *****
130 PRINT "DU ER I DIT ENMANDS-RUMSKIB"
140 PRINT "LØBET TØR FOR BRÆNDSTOF"
150 PRINT "UD FOR PLANETEN KARPATH."
160 PRINT "SITUATIONEN ER KRITISK,"
170 PRINT "DA DU HELLER IKKE"
180 PRINT "HAR RET MEGET ILT TILBAGE."
190 PRINT "DIN ENESTE CHANCE"
200 PRINT "FOR AT OVERLEVE"
210 PRINT "ER AT NA OVER"
220 PRINT "TIL ET ANDET RUMSKIB,"
230 PRINT "DER LIGESOM DIT EGET"
240 PRINT "LIGGER I KREDSLØB"
250 PRINT "HØJT OVER KARPATH."
260 GOSUB 4120
265 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
   CHR$(147) *****
270 PRINT "DU FORLADER DIT RUMSKIB"
280 PRINT "OG SVÆVER I DIN RUMDRAGT"
290 PRINT "OVER TIL DET FREMMEDE FARTØJ."

```

```

300 PRINT "DET SIDSTE, DU SER,"
310 PRINT "IDET DU SVÆVER IND"
320 PRINT "GENNEM RUMSKIBETS LUFTSLUSE,"
330 PRINT "ER PLANETEN KARPATHS MANE,"
340 PRINT "OG KNAK HAR LUFTSLUSEN"
350 PRINT "LUKKET SIG BAG DIG,"
360 PRINT "FØR DIN ILT SLIPPER OP."
370 PRINT "MEN DER ER ILT I SKIBET,"
380 PRINT "OG HVIS DU KAN FA DET STARTET,"
390 PRINT "KAN DU LANDE PA KARPATH.":GOSUB 4120

```

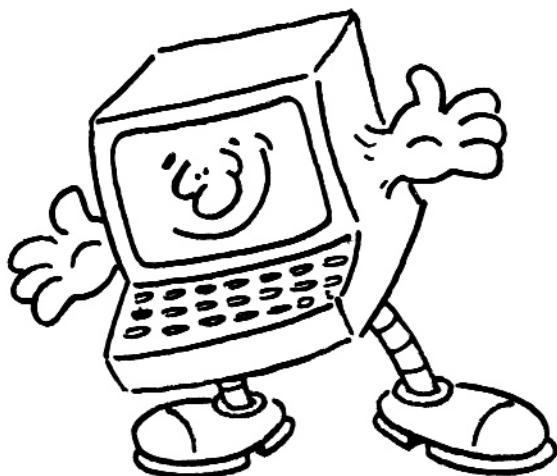
Bemærk, at jeg har lavet en linieafstand på 10. Hvis du ikke har en RENUM-facilitet, så vil det være klogt af dig under konstruktionen af et spil, der ikke er længere end dette, at bruge en større linieafstand, fx 100.

Vi tager sektionen linie for linie:

10-50: Vi rydder skærmen og laver en pæn lille overskrift til vores sensationelle spil.

60: En løkke, der laver 8 tomme linier.

70-110: Nogle praktiske bemærkninger til spilleren.



120: Vi går ned til mellemrumstangent-rutinen i linie 4120 (se denne) og rydder på ny skærmen.

130-390: Her kommer spillets introduktion – delt op med små hop ned til mellemrumstangent-rutinen i linie 4120 (se denne) og tilsvarende skærmrydninger, da vi ikke kan få plads til det hele på én gang. Introduktionen fortæller spilleren, hvad det hele går ud på og sætter ham (forhåbentlig) i den rette stemning. Bemærk, at vi allerede her giver ham et vigtigt fingerpeg ved at fortælle ham, at der er en måne uden for skibet.

Det navn, jeg har givet planeten – Karpath – er en lille spøg. Som man ved, er vampyrernes hjemland Transsilvanien, hvor man finder bjergkæden Karpaterne.

Vi går sporenstregs videre til næste sektion, der rummer vores DATA:

```
399 REM DATA *****
400 LO=2
410 AU=9:AN=23:AL=4:GL=20
420 DIM G(AN),G$(AN),U$(AU),N$(AN)
430 DIM V$(2),L$(AL),L(AL,2)
440 FOR I=1 TO GL:READ G(I):NEXT I
450 FOR I=1 TO GL:READ G$(I):NEXT I
460 FOR I=1 TO AU:READ U$(I):NEXT I
470 FOR I=1 TO AN:READ N$(I):NEXT I
480 FOR I=1 TO 2:READ V$(I):NEXT I
490 FOR I=1 TO AL:READ L$(I):NEXT I
500 FOR I=1 TO AL:FOR J=1 TO 2:READ L(I,J):
    NEXT J:NEXT I
510 DATA 0,1,2,2,0,0,2,0,0,2,2,3,0,0,3,2,1,0,1,0
520 DATA EN MANE,ET VINDUE,EN LUFTSLUSE,EN REOL,
    EN KORAN,ET LEKSIKON
530 DATA EN MUSEFÆLDE,EN ROTTE,ET TRÆSTYKKE
540 DATA EN HAMMER,EN KNIV,EN KISTE,EN VAMPIR,
    ET MAGNETKORT
```



```

550 DATA EN DØR,EN SIKRING,ET KRUCIFIKS,ET RUMSKIB,
    ET INSTRUMENTBRÆT,EN MAND
560 DATA GA,AKT,UND,BRU,LÆS,LÆG,GEM,FOR,TAG:REM HVIS
    Æ=AE SA STU I STEDET FOR LÆS *****
570 DATA MAN,VIN,LUF,REQ,KOR,LEK,MUS,ROT,TRÆ,HAM,
    KNI,KIS,VAM,MAG,DØR,SIK,KRU
580 DATA RUM,INS,MAN,SPI,FOR,AGT
590 DATA FORUD,AGTERUD
600 DATA COCKPIT,SALON,SOVEKABINE,MASKINRUM
610 DATA 0,2,1,3,2,0,3,0
620 GOSUB 630:GOSUB 3740:GOTO 650

```

Linie for linie:

400: Vi sætter LO til 2, hvilket betyder, at spillet begynder på Lokalitet nr. 2, rumskibets salon. Hvis du undrer dig over, hvorfor jeg ikke har medtaget de andre variabler (SK, MU etc.), er grunden den, at de ved spillets begyndelse har værdien 0 og derfor ikke behøver defineres. Visse computere accepterer ikke dette, så hvis du har en af dem, må du her i linie 400 også skrive de øvrige variabelnavne (se listen ovenfor) og gøre dem lig med nul.

410: Vi definerer antallet af udsagnsord, navneord, lokaliteter og genstande, der befinder sig på en lokalitet.

420-430: Vi DIMensionerer de DATAsæt, vi har brug for i spillet. G() skal bruges til oplysninger om, *hvor* genstandene befinder sig, G\$() skal bruges til at vise dem på skærmen som ord, U\$() skal fortælle maskinen, hvilke udsagnsord, der kan accepteres, og N\$() hvilke navneord, der kan accepteres. V\$() rummer de retningsangivelser, der skal skrives på skærmen (i dette tilfælde ikke verdenshjørner, men »forud« og »agterud«), L\$() rummer beskrivelsen af lokaliteterne og L() DATA for døre. Bemærk, at vi giver sættene G() og G\$() AN (23) elementer og ikke blot de 20 (GL), de tilhørende DATA-sætninger (i linie 510-550) indeholder. Det har ikke nogen praktisk betydning i *dette* spil, men er en god

generel forholdsregel at tage. Der er en nøje sammenhæng mellem indholdet af sættene G() og G\$() og indholdet af sættet N\$(), og N\$() indeholder AN elementer. Og hvis man skal have dem til at fungere sammen i fx en løkke, der hedder FOR I = 1 TO AN, er det vigtigt, der er afsat lige mange elementer til dem begge – selv om de sidste tre i G() og G\$() altså er »tomme«.

440-500: Vi indlæser (READ'er) de efterfølgende DATA. Bemærk, at selv om vi har givet sættene G() og G\$() AN (23) elementer, skal vi kun indlæse NL (20), hvilket svarer til det antal, der står i DATA-sætningerne . . . ellers vil der ske frygtelige ting og sager! Læs en nærmere udredning på problemet i kommentaren til forrige linie.

510: De DATA, der hører til sættet G(). Som du kan se, har nogle af dem værdien 0. Det betyder, at de – i hvert fald så længe de har værdien 0 – ikke vil kunne »ses« af spilleren. Nogle af dem vil få tildelt en ny værdi, når visse betingelser er opfyldt (fx roten, når den er gået i fælden), mens andre forbliver 0 hele spillet igennem (fx rumskibet, der jo ikke skal kunne »ses« liggende inde i sig selv). Vi er nødt til at tage dem med på trods af dette, da der (som allerede nævnt) kunne komme til at gå kludder i vores system, dersom indholdet af G() ikke nøje svarede til indholdet af G\$() og N\$(). Prøv selv at sammenligne disse tre sæts indhold og bemærk, at rækkefølgen *skal være den samme!* Dette er *meget vigtigt*. Hvis G\$(1) er EN MÅNE, *skal* N\$(1) være MÅN, og værdien af G(1) *skal* henvise til månen!

520-550: De data, der hører til sættet G\$(). Det er disse ord, der vil blive skrevet ud på skærmen – eller i hvert fald nogle af dem. Strengt taget kunne vi godt skrive KARTOFFELMOS eller blot X i stedet for fx RUMSKIB, der *aldrig* vil blive skrevet ud på skærmen, men for overskuelighedens skyld vil jeg råde dig til at tage disse »tomme« ord med alligevel, i hvert fald indtil du er dygtig nok til at kunne stryge dem uden at miste overblikket.

560: De udsagnsord, maskinen skal kunne genkende, kogt ned til de tre første bogstaver. Bemærk, at dersom din maskine ikke har

noget dansk »Æ«, så kan den ikke kende forskel på »LAEse« og »LAEgge«! Dette problem kan du fx løse ved at skrive STU (for STUdere) i stedet for LAE (for LAEse).

570-580: DATA for de navneord, maskinen skal kunne genkende, kognet ned til de tre første bogstaver. Hvis du ikke har dansk »Å« på din computer, så skriv MAA i stedet for MÅN – og TRA i stedet for TRÆ, hvis du mangler det danske »Æ«. Mangler du det danske »Ø«, kan du fx skrive OE eller bruge tegnet for nul, der på de fleste computere ligner det til forveksling. Eftersom dette er strengvariable, interesserer det ikke maskinen det fjerneste, at der står et tal i stedet for et bogstav inde i et ord, da den jo opfatter strengen som en *helhed*.

Bemærk også, at jeg her har medtaget de ekstra ord SPIL, FORud og AGTerud, som maskinen også skal kunne genkende. Det gør ikke noget, at »forud« og »agterud« ikke *er* navneord. Det har den dumme computer ikke nogen som helst anelse om. Den *vil* betragte dem som navneord, og det passer fint i vores kram.

590: DATA for retninger (dem, der vil blive skrevet ud på skærmen for at fortælle dig, hvor du kan gå hen).

600: Eftersom vi her opererer med kun fire lokalitetsbeskrivelser, kan vi godt skrive dem ind i en DATA-sætning uden at risikere at miste overblikket. Du husker nok, at jeg i gennemgangen af demonstrationsspillet slog fast med syvtommersøm, at vores L\$()·DATA burde opføres hver for sig – og det *bør* de også, hvis der er ret mange flere end disse fire. Men ingen regel uden undtagelse: her er der som sagt så få DATA at operere med, at vi lige så godt kan spare lidt plads i hukommelsen ved at opføre dem på denne måde. En anden væsentlig grund, til at vi kan gøre det, er at de ikke indeholder linieskift (se afsnittet »Skærmpræsentation« i næste kapitel).

610: DATA for døre. Det er disse DATA, der er uløseligt knyttet til dem i linien ovenfor, og normalt bør de stilles op som forklaret i gennemgangen af demonstrationsspillet. Men ... læs for-

klaringen til linie 600 (ovenfor) og forklaringen til demonstrationsspillet, hvis du ikke aner, hvad jeg taler om!

620: Vi går ned til 630 (se forklaring til denne), og derefter til linie 3740 (skærmtekst-rutinen) og endelig videre til linie 650 (strengdelings-rutinen).

Det var det. Lad os dernæst kigge på noget nyt – ændrings-rutinen:

```
629 REM ÆNDRINGER *****
630 IF DR=13 THEN DR=1
640 RETURN
```

I dette spil indeholder den kun én oplysning: den beder computeren kigge efter, om variabelen DR er lig 13 og i givet fald ændre den til 0. DR står for hvor mange grader rumskibet er blevet drejet om sin egen akse, og som du måske husker, skal tallet ganges med 30. Men da vi ikke kan have en cirkel på mere end 360 grader, beder vi computeren om at nulstille gradtallet, dersom vi er nået hele vejen rundt.

I det næste (og meget store) spil, jeg vil gennemgå (»På skattejagt i Amazonlandet«), er der mange ting, der skal checkes, hver eneste gang skærmen skal ryddes. Ved at lægge dette check på dette sted, sikrer vi os, at variabel-ændringer, der har fundet sted, *bliver* skrevet ud på skærmen (dvs. meddelt spilleren), når vi efter linie 640's RETURN vender tilbage til linie 620 og går ned til skærmtekst-rutinen.

Vi går videre:

```
649 REM STRENGDELING*****
650 PRINT:PRINT "HVAD VIL DU GØRE";
660 INPUT B$
670 PRINT
680 IF B$="R" THEN 620
```

```

690 IF B$="F" THEN B$="GA FORUD"
700 IF B$="A" THEN B$="GA AGTERUD"
710 IF B$="STATUS" THEN 2990
720 IF B$="STOP" THEN 3620
730 IF B$="HJÆLP" THEN 2680
740 FOR I=1 TO LEN(B$)
750 IF MID$(B$,I,1)=" " THEN 780
760 NEXT I
770 GOTO 4110:REM HVIS KUN EET ORD *****
780 UO$=LEFT$(B$,I-1)
790 NO$=RIGHT$(B$, (LEN(B$)-I))
800 IF UO$="GA" THEN 850:REM IKKE HVIS A=AA *****
810 IF LEN(UO$)<3 THEN 4110
820 UO$=LEFT$(UO$,3):FOR I=1 TO AU:IF UO$=U$(I) THEN
860
830 NEXT I
840 UO=0:GOTO 4110
850 UO=1:GOTO 870:REM IKKE HVIS A=AA *****
860 UO=I
870 IF LEN(NO$)<3 THEN 4110
880 NO$=LEFT$(NO$,3):FOR I=1 TO AN:IF NO$=N$(I) THEN
930
890 NEXT I
900 IF NO$="FLI" THEN NO=9:GOTO 940
910 IF NO$="SKO" THEN NO=2:GOTO 940
920 NO=0:GOTO 4110
930 NO=I
940 IF (UO<>4 AND NO<>10) AND RO=1 AND VA=0 THEN
GOSUB 2150:GOTO 3880

```

650-710: Se gennemgangen af demonstrationsspillet.

720: Vi giver her spilleren mulighed for at stoppe spillet ved at gå ned til linie 3620 (se denne).

730: I mange spil skal spilleren selv finde ud af, hvilke komman-

doer han skal bruge – men jeg mener, man skal spille med åbne kort, så jeg har lavet en hjælpe-rutine, der fortæller hvordan spillet skal spilles (og ikke spilles!), så den stakkels spiller ikke skal bruge flere timer på at afprøve diverse fantasifulde kommandoer (hvad siger du fx til »kysse vampyr«?) til ingen verdens nytte. Du kan finde hjælpe-rutinen i linie 2680 og fremefter.

740-890: Denne del af strengdelings-rutinen er allerede blevet gennemgået i demonstrationsspillet, så der er ingen grund til at forklare den yderligere.

900-930: De to første linier her var ikke med i demonstrations-spillet. De er strengt taget ikke nødvendige, og når jeg overhovedet har taget dem med, er det for at spare spilleren for et irritationsmoment. Et sted i spillet får han nemlig at vide, at han slår en flis af reolen med hammeren, og når han så prøver at TAGE FLIs, vil computeren fortælle ham, at det kan han ikke. Den reagerer nemlig på ordet TRÆstykke og har ikke nogen indbygget synonymordbog. Men her fortæller vi den, at ordet FLIs er det samme som ordet TRÆstykke, og det tager den så til efterretning. På lignende måde fortæller vi i linie 910, at SKOdde er lig med VINdue, hvilket ikke umiddelbart lyder logisk, men har sin forklaring i, at ordren AKTivere VINdue faktisk slår skodderne fra. Og da jeg er sikker på, at mange vil prøve at AKTivere



SKOdde, selv om computeren ikke accepterer ordet SKOdde, fortæller jeg den her, at det har den værsgo at gøre. Når du laver dine egne spil, er det på dette sted, du skal anføre synonymerne til de ord, du har anbragt i DATA-sættene, hvis du har en mistanke om, at de, der skal spille spillet, kunne finde på at bruge sådanne synonymer. I et andet spil i denne bog (»Mordet i den lukkede boks«), hvor jeg for eksempel har et RYGebord og en LÆNestol, har jeg fundet det rimeligt at bede computeren også at reagere på ordene BORD og STOL. Hvis der er flere forskellige borde og stole, bliver det straks mere kringlet. Hvordan dette problem løses, kan du se i »På skattejagt i Amazonlandet«, hvor der jongleres rundt med tre forskellige NØGler.

940: Hvis du gør noget som helst andet end at slå vampyren ihjel med hammeren og træstykket, når rotten er død, så dræber vampyren dig!

Vi snupper lige næste sektion:

949 REM UDSAGNSORD *****

**950 ON UO GOTO 960,1000,1270,1880,2340,2640,3050,
3310,3510**

Her er der kun linie 950 at forklare: ON GOTO findes ikke på alle computere, men er yderst praktisk. Den kæder noget, der normalt ville fylde en hel række linier, sammen i en. I stedet for den kunne man skrive:

**950 IF U0=1 THEN 960
951 IF U0=2 THEN 1000
952 IF U0=3 THEN 1270
953 IF U0=4 THEN 1880
954 IF U0=5 THEN 2340
955 IF U0=6 THEN 2640
956 IF U0=7 THEN 3050**

```

957 IF UO=8 THEN 3310
958 IF UO=9 THEN 3510

```

Det betyder, at hvis udsagnsordet (UO) er lig med 1 (dvs. hvis du har indtastet GÅ), skal maskinen gå ned til linie 960, hvis UO=2 (hvis du har indtastet AKTivere), skal maskinen gå ned til linie 1000 og så videre og så videre.

Vi tager næste sektion, det første af disse udsagnsord:

```

959 REM GA *****
960 IF NO<22 THEN 4110
970 NO=NO-21
980 IF L(LO,NO)=0 THEN PRINT "DU KAN IKKE GA DEN
    VEJ.":GOTO 650
990 LO=L(LO,NO):GOTO 620

```

960: Hvis man absolut skal spille Karl Smart og prøve at GÅ i andet end de to retninger, vi opererer med (hvis man fx skriver GÅ REOL, ryger maskinen straks ned til den linie, der fortæller, at det kan man godt spare sig. Den vil på skærmen skrive OM IGEN!

970-990: Vi har allerede i forrige linie elimineret de nederste 21 navneord (NO), og derfor kan vi i linie 970 med sindsro omnummerere de to sidste, FORud og AGTerud (Ja, jeg *ved* godt, det ikke er navneord, men det ved maskinen ikke!), så vi uden problemer kan bruge dem i linierne 980 og 990, hvor de skal passe til DATAsættet L(LO,NO), der ikke accepterer noget NO, der er højere end 2. Vi fortæller spilleren, at der er lukket for det varme vand, hvis han prøver at gå i en forbudt retning, og lader ham passere, dersom bevægelsen er tilladt. Han bliver flyttet forud eller agterud og vil allerede nu befinde sig i det næste rum, men det vil først blive skrevet ud på skærmen, når computeren har været igennem skærmtekst-subrutinen, hvortil den herfra bliver sendt via linie 620.

Vi går videre til udsagnsordet AKTivere, og nu begynder der at ske ting og sager, vi ikke har set før:

```
999 REM AKTIVERE *****
1000 IF NO=18 THEN GOSUB 1180:GOTO 650
1010 IF G(NO)=LO OR G(NO)=-1 THEN 1030
1020 PRINT "VÆR NU LIDT ALVORLIG, HVA'?:":GOTO 650
1030 IF NO<9 THEN 1060
1040 IF NO>8 AND NO<17 THEN 1070
1050 IF NO>16 THEN 1080
1060 ON NO GOSUB 1260,1120,1140,1260,1260,1260,1160,
      1260:GOTO 1090
1070 ON NO-8 GOSUB 1260,1260,1260,1260,1260,1260,
      1180,1260:GOTO 1090
1080 ON NO-16 GOSUB 1260,0,1190,1260,0,0,0
1090 IF SF=1 THEN 4030
1100 IF DE=1 THEN DE=0:GOTO 3880
1110 GOTO 650
1120 IF SK=1 THEN PRINT "SKODDERNE GLIDER TIL.":
      SK=0:RETURN
1130 PRINT "SKODDERNE GLIDER FRA.":SK=1:RETURN
1140 PRINT "LUFTEN FISER UD AF RUMSKIBET."
1150 PRINT "DU DØR NÆSTEN ØJEBLIKKELT.":DE=1:
      RETURN
1160 IF G(7)=-1 THEN PRINT "DU MA FØRST LÆGGE DEN
      FRA DIG.":RETURN
1170 PRINT "DU AKTIVERER MUSEFÆLDEN.":MU=1:RETURN
1180 PRINT "GOD IDE...MEN HVORDAN?":RETURN
1190 IF G(16)=4 AND TR<>2 THEN SF=1:RETURN:REM
      SF=SPIL FÆRDIGT *****
1200 IF TR=2 THEN 1220
1210 GOTO 1250
1220 PRINT "DET KAN DU IKKE.":PRINT "DU HAR SMADRET
      RAKETMOTOREN"
```

```

1230 PRINT "OG ØDELAGT DINE CHANCER":PRINT "FOR AT
      BLIVE FÆRDIG."
1240 PRINT "ET GODT RAD: BEGYND FORFRA.":RETURN
1250 PRINT "RUMSKIBET DREJER 30 GRADER.":DR=DR+1:
      GOSUB 630:RETURN
1260 PRINT "PRØV NOGET ANDET.":RETURN

```

1000: Se forklaring til linie 1080.

1010-1020: Hvis den genstand, man ønsker at aktivere, befinder sig på lokaliteten eller på ens person, så ned til 1030. Hvis ikke får man at vide, at man er dum i nakken.

1030-1080: Da jeg i denne bog holder mig til programlinier på højst 80 anslag (selv om der er datamater – min egen inklusive – der kan tage flere hundrede anslag pr. linie), bliver jeg nødt til at dele den ene ON GOSUB-linie, jeg egentlig kunne nøjes med, op i tre. Og i de tre første linier forgrener jeg navneordene (for det er dem, vi arbejder med her) i tre grupper: 1-8, 9-16 og 17-23, som derefter forgrenes videre i de tre næste linier. ON GOSUB svarer i princippet ganske til ON GOTO (se forklaringen til linie 950), blot med den forskel, at maskinen i sin gennemlæsning vender tilbage, før den går videre (præcis som i en GOSUB-linie). Efter udførelsen af subrutinen går den videre til linie 1090, hvilket det er nødvendigt at meddele den i de to øverste af ON GOSUB-linierne, men naturligvis ikke i den tredje, der står umiddelbart inden linie 1090. Hvis du studser over, hvorfor der i linie 1080 står 0 fire steder, er forklaringen den, at maskinen alligevel aldrig kommer til at læse disse nuller (hvis den gjorde det, ville den protestere mod, at der blev henvist til en linie, som ikke eksisterer), og når den ikke gør det, skyldes det, at de navneord, der knytter sig til disse positioner – RUMskib, SPil, FORud, AGTerud (ja, jeg *ved* godt, at kun RUMskib og SPil er rigtige navneord, men . . . nu gider jeg ikke forklare det mere!) – i forvejen er blevet siet fra i linie 1000 ved aldrig at kunne opfylde betingelserne her. De kan ikke befinde sig inden i rumskibet (et rum-

skib inden i et rumskib!), og de bliver heller ikke skrevet op som ting, man kan se. Men rumskibet burde faktisk kunne aktiveres – det er ikke fladpandet at skrive ordren AKTIVERE RUMSKIB – så derfor har jeg allerede henvist til svaret på dette i linie 1000, hvor jeg beder maskinen smutte ned til linie 1180, hvis der bliver bedt om dette. Jeg finder det et rimeligt forlangende (det er jo faktisk det, hele spillet går ud på), og et rimeligt forlangende kræver et mere høfligt svar end det, der ville være blevet skrevet ud i linie 1020, hvis henvisningen til linie 1180 ikke var føjet ind her.

1090-1100: Når man opererer med GOSUB-sætninger, må man aldrig henvise til nye linier, før subrutinen er blevet afsluttet med et RETURN (det vil i de fleste tilfælde opæde maskinens hukommelse med lynets hast), så derfor har vi her et par henvisninger, som ligger efter maskinens tilbagevenden til GOSUB-linien. Hvis SF=1 er spillet færdigt, og vi hopper ned til den lykkelige slutning, og hvis DE=1 er spilleren blevet slået ihjel, og vi hopper ned til den ulykkelige slutning. Måske vil du studse over, at disse to variabler ikke er medtaget i gemme/fortsætte-rutinen, men det er naturligvis fordi der ikke *er* noget at gemme eller fortsætte, hvis spillet er færdigt.

1110: Hvis du ikke er blevet slået ihjel eller har fuldendt spillet, går det videre. Tilbage til linie 650.

1120-1130: Hvad der sker, når man prøver at AKTivere VINduet (eller SKOdterne): hvis de er lukkede, glider de fra, og hvis de er åbne, glider de til. Elementært, kære Watson!

1140-1150: Hvis du AKTiverer LUFtslugen, dør du.

1160-1170: Hvad der sker, hvis man AKTiverer MUSefælden: Har man den på sig, kan rotten naturligvis ikke gå i den, og man får besked på at lægge den fra sig. Hvis den ligger på gulvet, tages ordren til efterretning. Fælden er nu aktiveret (MU=1).

1180: Hvis man forsøger at AKTivere DØRen eller RUMskibet, får man at vide, at det er en god idé, og nærmere instrukser udbedes.

1190-1250: Hvis sikringen – G(16) – er i Rum nr. 4 og man ikke har smadret raketmotoren (så er TR=2), er spillet færdigt, når man AKTiverer INStrumentbrættet. Er disse betingelser ikke opfyldt, får man enten at vide, at man har smadret motoren og godt kan begynde forfra, eller også drejer rumskibet 30 grader. Hvis det gør det, hopper maskinen en lille tur ned (eller rettere op) til linie 630 og undersøger, om DR=13 (dvs. om man er nået alle cirkelens 360 grader rundt og skal til at begynde på en ny omgang), hvorefter den fuldfører den første subrutine ved at RETURNere til linie 1080.

1260: Hvis du forsøger at AKTivere MÅNen, REOlen, KORanen, LEKsikonnet, ROTten, TRÆstykket, HAMmeren, KNIVEN, KISTen, VAMPyren, MAGnetkortet, SIKringen, KRUCifikset eller MANDen, får du besked på at prøve noget andet. Det er i øvrigt steder som dette, man kan højne oplevelsen ved spillet ved at anbringe små rappe bemærkninger. I stedet for det meget høflige PRØV NOGET ANDET kunne man lade computeren skrive bidske bemærkninger såsom TAG NU OG TÆNK DIG OM, HVA? eller NU ER JEG SNART TRÆT AF DIG eller HAR DU FLERE GENIALE FORSLAG? eller (hvis man vil være rigtig uhøflig) DU KAN RENDE MIG NOGET SÅ GRUSOMT.

Vi går videre med den sektion, der hører til udsagnsordet »undersøge«:

```
1269 REM UNDERSØGE *****
```

```
1270 IF NO=1 AND LO=1 THEN GOSUB 1400:GOTO 650:REM  
      MANE *****
```

```
1280 IF NO=14 AND G(13)=LO AND VA=0 THEN 1300
```

```
1290 GOTO 1320
```

```
1300 PRINT "DUM IDE."
```

```

1310 PRINT "VAMPYREN VAGNER OG SLAR DIG IHJEL.":GOTO
      3880
1320 IF NO=18 THEN PRINT "GOD IDE...BARE SE AT KOMME
      I GANG!":GOTO 650
1330 IF G(NO)<>LO AND G(NO)<>-1 THEN PRINT "MEGET
      MORSOMT.":GOTO 650
1340 IF NO<9 THEN 1370
1350 IF NO>8 AND NO<17 THEN 1380
1360 IF NO>16 THEN 1390
1370 ON NO GOSUB 0,1430,1470,1480,1550,1560,1570,
      1610:GOTO 650
1380 ON NO-8 GOSUB 1630,1650,1660,1670,1710,1740,
      1750,1770:GOTO 650
1390 ON NO-16 GOSUB 1780,0,1790,1860,0,0,0:GOTO 650
1400 IF SK=0 THEN 1430
1410 IF DR=3 THEN PRINT "DEN ER HALV.":RETURN
1420 PRINT "DU KAN IKKE SE DEN.":RETURN
1430 IF SK=0 THEN PRINT "SKODDERNE ER LUKKEDE.":
      RETURN
1440 PRINT "SKODDERNE ER ÅBNE."
1450 IF DR=3 THEN PRINT "DU KAN SE EN MANE UDENFOR."
1460 RETURN
1470 PRINT "DEN ER LUKKET.":RETURN
1480 PRINT "I REOLEN, DER ER AF FYRRETRÆ,":PRINT "ER
      DER ";
1490 IF (G(5)=0 AND G(6)=0) OR (G(5)=2 AND G(6)=2)
      THEN 1510
1500 GOTO 1520
1510 PRINT "EN KORAN OG ET LEKSIKON.":G(5)=2:G(6)=2:
      RETURN
1520 IF G(5)=2 AND G(6)<>2 THEN PRINT "EN KORAN.":
      RETURN
1530 IF G(5)<>2 AND G(6)=2 THEN PRINT "ET
      LEKSIKON.":RETURN
1540 PRINT "INGENTING.":RETURN

```

```

1550 PRINT "DER STAR NOGET I DEN.":RETURN
1560 PRINT "DER STAR NOGET I DET.":RETURN
1570 IF MU>0 THEN PRINT "DEN ER AKTIVERET.":MU=MU+1
1580 IF MU>3 THEN PRINT "DER SIDDER EN STOR,":PRINT
      "FED ROTTE I DEN.":G(8)=G(7)
1590 IF MU=0 THEN PRINT "DEN ER IKKE AKTIVERET."
1600 RETURN
1610 IF RO=1 THEN PRINT "DEN ER DØD.":RETURN
1620 PRINT "DEN LEVER ENDNU.":RETURN
1630 IF TR>0 THEN PRINT "DET ER SPIDST":RETURN
1640 PRINT "DET ER STUMPT":RETURN
1650 PRINT "DET ER EN GANSKE ALMINDELIG HAMMER.":
      RETURN
1660 PRINT "DET ER EN GANSKE ALMINDELIG KNIV":RETURN
1670 PRINT "DEN ER AF ALUMINIUM."
1680 IF VA=1 THEN PRINT "DEN ER TOM.":RETURN
1690 IF G(20)=0 THEN PRINT "DER LIGGER EN MAND I
      DEN.":G(20)=LO
1700 RETURN
1710 PRINT "HANS BID ER DRÆBENDE."
1720 IF G(14)=0 THEN PRINT "HAN HAR ET MAGNETKORT I
      BRYSTLOMMEN."
1730 RETURN
1740 PRINT "DET KAN ÅBNE DØRE.":RETURN
1750 IF DO=1 THEN PRINT "DEN ER ÅBEN.":RETURN
1760 PRINT "DEN ER LAST.":RETURN
1770 PRINT "DEN ER TIL EN RAKETMOTOR.":RETURN
1780 PRINT "DET ER AF GULD OG BESAT MED ÆDELSTEN.":
      RETURN
1790 IF G(16)=4 AND TR<>2 THEN 1810
1800 GOTO 1820
1810 PRINT "EN GRØN LAMPE VISER,":PRINT "AT
      HOVEDMOTOREN NU FUNGERER.":RETURN
1820 PRINT "EN RØD LAMPE VISER,":PRINT "AT DER ER
      NOGET I VEJEN"

```

```

1830 PRINT "MED HOVED-RAKETMOTOREN.":PRINT "DET
      ENESTE, DER FUNGERER, "
1840 PRINT "ER DE SMA DYSE, DER":PRINT "KAN FA
      RUMSKIBET TIL AT"
1850 PRINT "DREJE OM SIN EGEN AKSE.":RETURN
1860 PRINT "HAN ER VAMPYR.":G(13)=LO:G(20)=0:GOTO
      1710

```

Linie for linie:

1270: her tager vi højde for Navneord nr. 1 (månen), der ville resultere i en kvik bemærkning i linie 1330, hvis vi ikke gjorde det.

1280-1310: Her gælder det samme for Navneord nr. 14 (magnetkortet), bortset fra at vi ikke laver en subrutine, men ordner sagen med det samme. Hvis $VA=0$ (hvis vampyren ikke er død), bliver man slået ihjel af ham, dersom man forsøger at undersøge det magnetkort, han har i brystlommen.

1320: Hvis man forsøger at UNDERsøge RUMskibet, får man at vide, at det er en god idé.

1330: Her frasorteres resten af de ting, der ikke befinder sig på stedet (eller rettere: på skærmen) eller på spillerens person.

1340-1390: Samme fremgangsmåde som under »aktivere«.

1400-1420: Hvad der sker, hvis man forsøger at UNDERsøge MÅNen. Hvis skodderne er lukkede, får man dette at vide. Hvis de er åbne, og man har drejet rumskibet 90 grader, får man at vide, at månen er halv – ellers at man ikke kan se den.

1430-1460: Hvis man UNDERsøger VINduet, er skodderne enten åbne eller lukkede. Hvis $DR=3$, kan man se en måne udenfor.

1470: Hvis man UNDERsøger LUFtslugen, får man at vide, at den er lukket (for hvis man prøver at åbne den, er spillet forbi. R.I.P.)

1480-1540: Hvad der sker, hvis man UNDERsøger REOlen.

1550-1560: Hvis man UNDERsøger henholdsvis KORanen og LEKsikonnet, får man (logisk nok) at vide, at der står noget i dem.

1580-1600: De forskellige svar på »UNDERsøge MUSEfælden«.

1610-1620: Svar på »UNDERsøge ROTte«.

1630-1640: Svar på »UNDERsøge TRÆstykke«.

1650-1660: Svar på »UNDERsøge KNIV« og »UNDERsøge HAMmer«.

1670-1700: Svar på »UNDERsøge KISTen«.



1710-1730: Svar på »UNDERsøge VAMPyren«.

1740: Svar på »UNDERsøge MAGnetkortet«.

1750-1760: Svar på »UNDERsøge DØRen«.

1770-1780: Hvad man får at vide, hvis man UNDERsøger henholdsvis SIKringen og KRUCifikset.

1790-1850: Svar på forespørgslen »UNDERsøge INSTRumentbræt-tet«.

1860: Svar på »UNDERsøge MANDen«.

Vi tager næste del af programmet:

```
1879 REM BRUGE *****
1880 IF G(NO)<>LO AND G(NO)<>-1 THEN GOSUB 2330:GOTO
      650
1890 IF NO<9 THEN 1920
1900 IF NO>8 AND NO<17 THEN 1930
1910 IF NO>16 THEN 1940
1920 ON NO GOSUB 0,1970,1970,1970,1970,1970,1970,
      1970:GOTO 1950
1930 ON NO-8 GOSUB 1970,1980,2180,1970,1970,2280,
      1970,2300:GOTO 1950
1940 ON NO-16 GOSUB 2320,1970,1970,1970,0,0,0
1950 IF DE=1 THEN DE=0:GOTO 3880
1960 GOTO 650
1970 PRINT "DU MÅ UDTRYKKE DIG LIDT MERE PRÆCIST.":
      RETURN
1980 ON LO GOSUB 1990,2110,2150,2160:RETURN:REM
      HAMMER *****
1990 IF (G(9)=-1 OR G(9)=LO) AND SK=1 AND VA=0 AND
      DR=3 AND G(13)=LO THEN 2010
2000 GOTO 2030
2010 PRINT "DU DRÆBER VAMPYREN":PRINT "MED HAMMEREN
      OG TRÆSTYKKET."
2020 G(14)=LO:VA=1:G$(13)="EN DØD VAMPYR":RETURN
2030 IF DR<>3 AND VA=0 AND G(13)=LO THEN 2050
2040 GOTO 2080
```

```

2050 PRINT "DU HAR IKKE FUNDET UD AF":PRINT "AT
      SVÆKKE VAMPYREN,"
2060 PRINT "OG DERFOR KAN HAN UDEN BESVÆR":PRINT
      "OVERMANDE OG DRÆBE DIG."
2070 DE=1:RETURN
2080 IF G(13)<>LO OR VA=1 THEN PRINT "DU SMADRER
      VINDUET,"
2090 PRINT "OG LUFTEN SIVER UD AF":PRINT "RUMSKIBET.
      DU DØR AF LUFTMANGEL."
2100 DE=1:RETURN
2110 IF G(9)=0 THEN PRINT "DU SLAR EN FLIS AF
      REOLEN.":G(9)=LO:RETURN
2120 IF G(13)=LO THEN 2150
2130 PRINT "DU SMADRER LUFTSLUSEN,":PRINT "OG LUFTEN
      SIVER UD AF"
2140 PRINT "RUMSKIBET. DU DØR AF LUFTMANGEL.":DE=1:
      RETURN
2150 PRINT "DU BLIVER DRÆBT AF VAMPYREN.":DE=1:
      RETURN
2160 PRINT "DU SMADRER RAKETMOTOREN,":PRINT "HVORFOR
      GJORDE DU NU OGSÅ DET?"
2170 PRINT "NU ER DU NØDT TIL":PRINT "AT BEGYNDE
      FORFRA.":TR=2:RETURN
2180 IF (G(9)=LO OR G(9)=-1) AND TR=0 THEN 2200
2190 GOTO 2210
2200 PRINT "DU SPIDSER TRÆSTYKKET.":TR=1:RETURN
2210 IF (G(8)=LO OR G(8)=-1) AND RO=0 THEN 2230
2220 GOTO 2260
2230 RO=1:PRINT "DU DRÆBER ROTTEN.":PRINT "DENS BLOD
      SPRØJTER UD PÅ GULVET."
2240 PRINT "LUGTEN VÆKKER VAMPYREN,":PRINT "DER
      KOMMER FARENDE"
2250 PRINT "OG GAR TIL ANGREB PÅ DIG.":G(13)=LO:
      RETURN
2260 PRINT "DU SKÆRER DIG SELV I HANDLEDDET":PRINT

```

```

      "...OG FORBLØDER."
2270 DE=1:RETURN
2280 IF LO=1 OR LO=2 THEN PRINT "DET FAR DU IKKE
      NOGET UD AF...HER.":RETURN
2290 PRINT "DU ÅBNER DØREN.":L(3,2)=4:RETURN
2300 IF LO<>4 THEN PRINT "DET FAR DU IKKE NOGET UD
      AF...HER.":RETURN
2310 PRINT "DU UDSKIFTER SIKRINGEN.":G(16)=4:RETURN
2320 PRINT "DET FAR DU IKKE NOGET UD AF.":RETURN
2330 PRINT "PRØV NOGET ANDET, FISTER.":RETURN
2339 REM LÆSE *****
2340 IF G(NO)<>LO AND G(NO)<>-1 THEN 2360
2350 GOTO 2370
2360 PRINT "IH, HVOR ER DU ALTSA MORSOM.":GOTO 650
2370 IF NO<>5 THEN 2480
2380 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
2385 PRINT "FORREST I KORANEN STAR SKREVET:"
2390 PRINT:PRINT "'DENNE HELLIGE BOG GIVER VI DIG,"
2400 PRINT "VOR ELSKEDE SØN, VLAD NOSFERATU,"
2410 PRINT "HER PÅ DIN 10-ARS FØDSELS DAG,"
2420 PRINT "AT DU KAN LÆSE ALLAHS ORD"
2430 PRINT "OG BLIVE HANS TRO TJENER."
2440 PRINT "MA DEN HELLIGE HALVMANES"
2450 PRINT "FORBANDELSE RAMME ENHVER VANTRO"
2460 PRINT "OG ENHVER TROENDE, DER ER"
2470 PRINT "BLEVET FORLEDT AF SATAN!":GOTO 650
2480 IF NO<>6 THEN 2610
2490 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
2495 PRINT "I LEKSIKONNET STAR BLANDT ANDET:"
2500 PRINT:PRINT "'VAMPYRER:"
2510 PRINT "SKAL MAN USKADELIGGØRE EN VAMPYR,"
2520 PRINT "MA MAN GENNEBØRE HANS HJERTE"
2530 PRINT "MED ET SPIDST STYKKE TRÆ."

```

```

2540 PRINT "FØRST MA MAN DOG SVÆKKE VAMPYREN,"
2550 PRINT "HVIS KRÆFTER ER OVERMENNESKELIGE,"
2560 PRINT "MED ET HELLIGT SYMBOL, SASOM ET"
2570 PRINT "KORS ELLER EN DAVIDSSTJERNE E.L."
2580 PRINT "- AFHÆNGIGT AF, HVILKEN RELIGION"
2590 PRINT "HAN BEKENDTE SIG TIL, FØR HAN"
2600 PRINT "BLEV EN AF SATANS TJENERE.":GOTO 650
2610 IF NO<>14 THEN 2630
2620 PRINT "DER STAR 'MADE IN TRANSYLVANIA' PA
      DET.":GOTO 650
2630 PRINT "DET FAR DU IKKE NOGET UD AF.":GOTO 650
2639 REM LÆGGE *****
2640 IF G(NO)<>-1 THEN 2660
2650 GOTO 2670
2660 IF G(NO)=-1 THEN 2670
2665 PRINT "FOR AT KUNNE LÆGGE NOGET":PRINT "MA MAN
      HAVE DET.":GOTO 650
2670 PRINT "SOM DU VIL.":N=1:GOSUB 4040:G(NO)=LO:
      GOTO 620

```

På indeværende tidspunkt burde du være tilstrækkelig meget inde i systemet til, at vi kan hoppe videre uden yderligere forklaringer. Blot lige én ting: Det lille »N« i linie 2670 er en variabel, vi indfører for at spare plads. Den skal bruges i løkke-subrutinen (se denne i linie 4060), hvor den ganges med 1000 og altså er med til at bestemme, hvor lang en pause maskinen skal holde på dette og andre steder. Når vi overhovedet har denne »tomme« løkke-subrutine med, er det for at vi skal kunne nå at læse teksten (i dette tilfælde »UDMÆRKET«), inden skærmen bliver rydlet. Nogle maskiner er hurtigere til at køre løkker igennem end andre, og hvis du synes teksten på steder som dette står for kort tid, inden den forsvinder, er det nok fordi din maskine er hurtigere end den, jeg har brugt. I så fald kan du meget nemt få teksten til at stå længere ved at forhøje dette »N«.

Næste sektion er hjælpe-rutinen, der sådan set også taler for sig selv:

```
2679 REM HJÆLP *****
2680 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
2685 PRINT "DU KAN BRUGE"
2690 PRINT "FØLGENDE KOMMANDOER:"
2700 PRINT:PRINT "GA (FORUD OG AGTERUD)"
2710 PRINT "_ MEN HER KAN DU NØJES MED"
2720 PRINT "AT SKRIVE 'F' FOR 'GA FORUD'"
2730 PRINT "OG 'A' FOR 'GA AGTERUD'."
2740 PRINT:PRINT "TAGE _ LÆGGE _ BRUGE _ LÆSE":REM
      HVIS LAE SA STU *****
2750 PRINT "_ AKTIVERE _ UNDERSØGE."
2760 PRINT "(DISSE ORD SKAL EFTERFØLGES AF"
2770 PRINT "NAVNEORD _ F.EKS: 'TAGE HAMMER')."
```

```

2910 PRINT "STATUS: VISER HVAD DU BÆRER."
2920 PRINT:PRINT "ET SIDSTE LILLE TIP:"
2930 PRINT "DU KAN VED AT SKRIVE"
2940 PRINT "'R' RYDDE SKÆRMEN"
2950 PRINT "OG SE, HVOR DU ER"
2960 PRINT "(HVIS STEDSANGIVELSEN"
2970 PRINT "FOROVEN ER FORSVUNDET)."
2980 GOSUB 4120:GOTO 620

```

Også status-rutinen kender du fra demonstrationsspillet:

```

2989 REM STATUS *****
2990 PRINT "DU BÆRER FØLGENDE:"
3000 BP=0
3010 FOR I=1 TO AN:IF G(I)=-1 THEN PRINT G$(I):BP=1
3020 NEXT I
3030 IF BP=0 THEN PRINT "INGENTING"
3040 GOTO 650

```

Og nu over til noget helt nyt:

```

3049 REM GEMME SPIL *****
3050 IF NO=21 THEN 3080:REM HAR DU KASSETTEBAND-
      STATION, SA SE FORKLARINGEN TIL DENNE LINIE!
      *****
3060 PRINT "DET ENESTE, DU KAN GEMME,":PRINT "ER
      SPILLET.":GOTO 650
3080 OPEN "O",1,"RUM-DATA":REM COMMODORE-BRUGER,
      SKRIV I STEDET: OPEN 1,1,1,"RUM-DATA" *****
3090 PRINT# 1,SK
3100 PRINT# 1,MU
3110 PRINT# 1,TR

```

```

3120 PRINT# 1,DO
3130 PRINT# 1,DR
3140 PRINT# 1,RO
3150 PRINT# 1,VA
3160 PRINT# 1,LO
3170 FOR I=1 TO GL:PRINT# 1,G(I):NEXT I
3180 FOR I=1 TO GL:PRINT# 1,G*(I):NEXT I
3190 FOR I=1 TO AU:PRINT# 1,U*(I):NEXT I
3200 FOR I=1 TO AN:PRINT# 1,N*(I):NEXT I
3210 FOR I=1 TO 2:PRINT# 1,V*(I):NEXT I
3220 FOR I=1 TO AL:PRINT# 1,L*(I):NEXT I
3230 FOR I=1 TO AL:FOR J=1 TO 2:PRINT# 1,L(I,J):NEXT
      J:NEXT I
3240 CLOSE 1
3250 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
3255 PRINT "ØNSKER DU AT FORTSÆTTE SPILLET?"
3260 GOSUB 4080
3270 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I
      STEDET: GET A$
3275 IF A$="" THEN 3270
3280 IF A$="J" THEN 3340
3290 IF A$="N" THEN 4100
3300 GOTO 3270
3309 REM FORTSÆTTE SPIL *****
3310 IF NO=21 THEN 3340
3320 PRINT "DET ENESTE,":PRINT "DU KAN FORTSÆTTE, ER
      SPILLET.":GOTO 450
3340 OPEN "I",1,"RUM-DATA":REM COMMODORE-BRUGER,
      SKRIV I STEDET: OPEN 1,1,0,"RUM-DATA" *****
3350 INPUT# 1,SK
3360 INPUT# 1,MU
3370 INPUT# 1,TR

```

```

3380 INPUT# 1,DO
3390 INPUT# 1,DR
3400 INPUT# 1,RO
3410 INPUT# 1,VA
3420 INPUT# 1,LO
3430 FOR I=1 TO GL:INPUT# 1,G(I):NEXT I
3440 FOR I=1 TO GL:INPUT# 1,G$(I):NEXT I
3450 FOR I=1 TO AU:INPUT# 1,U$(I):NEXT I
3460 FOR I=1 TO AN:INPUT# 1,N$(I):NEXT I
3470 FOR I=1 TO 2:INPUT# 1,V$(I):NEXT I
3480 FOR I=1 TO AL: INPUT# 1,L$(I):NEXT I
3490 FOR I=1 TO AL:FOR J=1 TO 2:INPUT# 1,L(I,J):NEXT
      J:NEXT I
3500 CLOSE:GOTO 620

```

Linie 3049-3500: Her gemmer og fortsætter vi vores spil. Det er som nævnt uhyre praktisk at kunne gøre dette, hvis man er i gang med et langt og meget indviklet spil og bliver nødt til at afbryde det uden at være færdig. Det er også en fordel at kunne gemme spillet på et sted, hvor man ikke rigtig ved, hvad der vil ske, hvis man prøver noget nyt. Hvis man fx for 117. gang prøver at tage magnetkortet fra vampyren og er blevet træt af at blive slået ihjel hver gang, kan man ved at gemme spillet fortsætte det fra det sted, hvor man gjorde noget galt. Det har måske ikke den store betydning i dette lille spil, som man hurtigt kan løbe igennem fra begyndelsen, men tro mig – er spillet blot en smule større, som fx det næste i denne bog, kan det spare en for mange irritationer at kunne gemme det på de kritiske steder.

Fremgangsmåden er uhyre enkel. Man gemmer simpelt hen de forskellige variablers værdi på det sted i spillet, man er kommet til. Hvis fx RO=1, ved maskinen at rotten er død, og hvis variablen RO gemmes med værdien 1, vil den stadig være død, når spillet kaldes frem igen. Men hvordan gør vi så det?

Jo, vi opretter en fil, der gemmer variablernes værdi. Lad os tage det linie for linie:

3050-3060: Hvis navneordet er andet end »SPIL«, får vi en vrissen

bemærkning og bliver gennet tilbage til udgangspunktet. Hvis du har en computer med kassettebåndstation og gerne vil have en ekstra lille påmindelse, så bemærk, at du i linie 3050 skal skrive »THEN 3070« i stedet for »THEN 3080« – og tilføj linie 3070, der hedder:

```
3070 PRINT "LÆG ET BAND I MASKINEN.":PRINT
```

Hvis du derimod har diskettstation (eller er ligeglad med påmindelsen), skriver du bare programmet af, som det er LISTet.

3080: Husk, at denne linie er skrevet i Microsoft BASIC. På fx en Commodore vil den hedde

```
3080 OPEN 1,1,1,"RUM-DATA"
```

hvilket er en anelse anderledes end Microsoft-versionen. Check manualen til din computer, hvis du løber ind i problemer i dette afsnit.

3090-3160: Her gemmes vores ikke-indekserede variabler. Mange computere, fx IBM PC, ville acceptere disse linier samlet i én, men hvis du skriver dem af, som de står her, skulle der ikke kunne ske noget galt (i hvert fald ikke, hvis du har en Commodore). Husk det lille »1,« foran variabelnavnet. Det er obligatorisk og fortæller computeren, at filen har nummeret 1. Hvis du i de spil, du laver, har brug for flere filer, så sørg for, at nummeret på filen stemmer overens med det nummer, du skriver foran de enkelte variabler.

3170-3230: Her gemmes de indekserede variabler. Fremgangsmåden er den samme som i READ-sætningerne, blot skal vi her bruge PRINT # i stedet for READ (og husk endelig det lille »1,« foran variabelnavnet).

3240: Vi lukker filen (husk altid dette CLOSE!)

3250-3300: Maskinen spørger dig, om du ønsker at fortsætte spillet og reagerer på dit svar. Bemærk, at instruktionen A\$=IN-KEY\$ hedder GET A\$ på Commodore BASIC.

3309-3500: Vi henter spillet (dvs. de gemte variabelværdier) frem og fortsætter det. Fremgangsmåden er faktisk nøjagtig den samme som i gemme-rutinen, blot bruger vi instruktionen INPUT# i stedet for PRINT#. Bemærk, at der også er variationer i linie 3340, som i Microsoft-versionen kræver et »I«, når der skal INPUT#tes. Bruger din computer andet en Microsoft BASIC, så kig i din computer-manual.

Hvis du har kassettebåndstation, kan du i linie 3310 skrive »3330« i stedet for »3340« og indføre en ny linie 3330, hvor du skriver:

```
3330 PRINT "FIND FILEN PÅ BANDET."
```

Resten af spillet burde sådan set også tale for sig selv, så vi nøjes med at bringe LISTningen:

```
3509 REM TAGE *****
3510 IF NO=14 AND VA=0 AND G(13)=LO THEN 3530
3520 GOTO 3550
3530 PRINT "DET SKULLE DU IKKE HAVE GJORT."
3540 PRINT "VAMPYREN SLAR DIG IHJEL.":GOTO 3880
3550 IF NO<5 OR NO=12 OR NO=13 OR NO=15 OR NO>17
    THEN NO=0
3560 IF NO=0 THEN PRINT "DET KAN DU IKKE.":GOTO 650
3570 IF G(NO)=-1 THEN PRINT "DET HAR DU ALLEREDE
    GJORT.":GOTO 650
3580 IF G(NO)<>LO THEN PRINT "MEGET MORSOMT.":GOTO
    650
```

```

3590 IF NO=7 AND MU>3 THEN G(8)=-1
3600 IF NO=8 OR NO=7 THEN MU=0
3610 G(NO)=-1:PRINT "UDMÆRKET." :N=1:GOSUB 4060:GOTO
    620
3619 REM STOPPE *****
3620 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****
3625 PRINT "DU ØNSKER AT STOPPE SPILLET?"
3630 GOSUB 4080
3640 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I
    STEDET: GET A$ *****
3645 IF A$="" THEN 3640
3650 IF A$="J" THEN 3680
3660 IF A$="N" THEN 650
3670 GOTO 3620
3680 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****
3685 PRINT "ØNSKER DU AT GEMME SPILLET?"
3690 GOSUB 4080
3700 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I
    STEDET: GET A$ *****
3705 IF A$="" THEN 3700
3710 IF A$="J" THEN 3080
3720 IF A$="N" THEN 4100
3730 GOTO 3680
3739 REM SKÆRMTEKST-RUTINE *****
3740 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****
3750 DU$="DU ER I RUMSKIBETS "
3760 PRINT DU$;L$(LO)
3770 FOR I=1 TO GL:IF G(I)=LO THEN SE=1
3780 NEXT I
3790 IF SE=0 THEN PRINT:GOTO 3840
3800 PRINT:PRINT "DU KAN SE: "
3810 FOR I=1 TO GL:IF G(I)=LO THEN PRINT G$(I):SE=0

```

```

3820 NEXT I:PRINT:GOTO 3840
3830 NEXT I
3840 PRINT "DU KAN GA:"
3850 FOR I=1 TO 2:IF L(LO,I)<>0 THEN PRINT V$(I)
3860 NEXT I
3870 RETURN
3879 REM AFSLUTNINGS-RUTINE *****
3880 PRINT:PRINT "BEKLAGER...SADAN ER LIVET."
3890 PRINT:PRINT "HAR DU LYST TIL":PRINT "AT PRØVE
      EN GANG TIL?"
3900 GOSUB 4080
3910 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I
      STEDET: GET A$ *****
3915 IF A$="" THEN 3910
3920 IF A$="J" THEN 3950
3930 IF A$="N" THEN 4100
3940 GOTO 3890
3950 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
3955 PRINT "HVIS DU HAR GEMT SPILLET":PRINT "OG
      ØNSKER AT FORTSÆTTE,"
3960 PRINT "SA TRYK PÅ 'F' FOR 'FORTSÆT'."
3970 PRINT "HVIS DU VIL BEGYNDE HELT FORFRA,"
3980 PRINT "SA TRYK PÅ 'B' FOR":PRINT "'BEDRE HELD
      NÆSTE GANG'."
3990 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I
      STEDET: GET A$ *****
3995 IF A$="" THEN 3990
4000 IF A$="F" THEN 3340
4010 IF A$="B" THEN RUN
4020 GOTO 3990
4030 PRINT "DU STARTER RUMSKIBETS HOVEDMOTOR"
4040 PRINT "OG LANDER SIKKERT PÅ KARPATH."
4050 PRINT "DU HAR GENNEMFØRT SPILLET.":PRINT:PRINT
      "TIL LYKKE!":END

```

```
4059 REM LØKKE-SUBROUTINE *****
4060 FOR X=1 TO (N*1000):NEXT X
4070 RETURN
4079 REM JA/NEJ-SUBROUTINE *****
4080 PRINT:PRINT "TRYK PÅ 'J' FOR 'JA'":PRINT "ELLER
      'N' FOR 'NEJ'."
4090 RETURN
4100 END
4109 REM OM IGEN-RUTINE *****
4110 PRINT "OM IGEN!":GOTO 650
4119 REM MELLEMRUMSTANGENT-RUTINE *****
4120 PRINT:PRINT "TRYK PÅ MELLEMRUMSTANGENTEN,"
4130 PRINT "NÅR DU VIL FORTSÆTTE."
4140 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I
      STEDET: GET A$ *****
4145 IF A$="" THEN 4140
4150 IF A$(">") THEN 4140
4160 RETURN
```

4 · Nogle videre finesser



Når du er kommet så langt som hertil, ved du, hvordan man laver et eventyrlandskab med genstande, som kan tages, lægges og på forskellig vis manipuleres. Du har forhåbentlig også en idé om, hvordan man opbygger en handling – men inden du kaster dig ud i selv at lave dine eventyr, har du måske lyst til at få nogle flere tips – eller måske har du allerede prøvet at lave nogle eventyr og er moden til at gå et stykke videre.

Godt – så lad os gå et stykke videre:

Skærmpresentation

Med »skærmpresentation« mener jeg det tekstbillede, du får frem, når din computer PRINTer instrukserne ud. Lad os tage et tilfældigt eksempel:

DU STAR I ET GAMMELT TEMPEL. SOLENS LYS FALDER I EN
STRIBE NED GENNEM ET HUL I LOFTET OG RAMMER ET
GADEFULDT RELIEF PÅ DEN FUGTIGE MOSTILGROEDE VÆG.

Nogle computere har en skærbredde på 40 anslag, andre på 80, andre igen et helt tredje tal. Hvis man fx har en skærbredde på 32 anslag, vil ovenstående tekst komme til at se sådan ud:

DU STAR I ET GAMMELT TEMPEL. SOL
ENS LYS FALDER I EN STRIBE NED G
ENNEM ET HUL I LOFTET OG RAMMER
ET GADEFULDT RELIEF PÅ DEN FUGTI
GE MOSTILGROEDE VÆG.

Det er ikke særlig pænt. Ordene er delt tilfældigt af computeren, og vi har svært ved at læse dem. Vi kunne løse dette problem ved at indføre en ny programlinie (eller i hvert fald et nyt PRINT), hver gang vi ønsker, der skal skiftes linie på skærmen:

```
10 PRINT "DU STAR I ET GAMMELT TEMPEL."  
20 PRINT "SOLENS LYS FALDER I EN STRIBE"  
30 PRINT "NED GENNEM ET HUL I LOFTET OG"  
40 PRINT "RAMMER ET GADEFULDT RELIEF PÅ"  
50 PRINT "DEN FUGTIGE MOSTILGROEDE VÆG."
```

På skærmen ville komme til at stå:

DU STAR I ET GAMMELT TEMPEL.
SOLENS LYS FALDER I EN STRIBE
NED GENNEM ET HUL I LOFTET OG
RAMMER ET GADEFULDT RELIEF PÅ
DEN FUGTIGE MOSTILGROEDE VÆG.

Det var straks meget bedre. Problemet er løst – så længe vi opererer med ting, vi *direkte* kan bede computeren om at PRINTe ud på skærmen. Men hvad nu hvis vi ønsker at gemme denne sætning som en del af et sæt? Vi har, som du husker, vores lokalitetsbeskrivelser samlet i det indekserede variabelsæt L\$(). Hvordan kan vi få computeren til at skifte linie de rigtige steder, hvis vi fortæller den, at fx L\$(20)=»DU STÅR I ET GAMMELT TEMPEL. SOLENS LYS FALDER I EN STRIBE NED GENNEM ET HUL I LOFTET OG RAMMER ET GÅDEFULDT RELIEF PÅ DEN FUGTIGE MOSTILGROEDE VÆG«?

Der melder sig straks flere problemer. Ikke alle computere kan have så lang en sætning i én linie, men det er der råd for:

```
10 L$1="I EN STRIBE NED GENNEM ET HUL I LOFTET OG
    RAMMER"
20 L$2=" ET GADEFULDT RELIEF PÅ DEN FUGTIGE
    MOSTILGROEDE VÆG."
30 L$(20)="DU STAR I ET GAMMELT TEMPEL. SOLENS LYS
    FALDER "+L1$+L$2
```

Hvis vi derefter i en linie 40 beder computeren PRINTe L\$(20), vil den (dersom vi stadig opererer med en skærmbredde på 32) PRINTe

```
DU STAR I ET GAMMELT TEMPEL. SOL
ENS LYS FALDER I EN STRIBE NED G
ENNEM ET HUL I LOFTET OG RAMMER
ET GADEFULDT RELIEF PÅ DEN FUGTI
GE MOSTILGROEDE VÆG.
```

Altså præcis det samme, som vi så for lidt siden. Men ét problem løste vi da: hvordan vi kan gemme en lang sætning i én variabel, selv om vi skal bruge flere linier til det. Men når vi gemmer en sådan sætning i en variabel, kan vi ikke putte *linieskift* ind i den.

Eller kan vi? Lad os først kigge på den løsning, man hyppigst ser brugt (stadig til en maskine med en skærbredde på 32):

```
10 L$(20)="DU STAR I ET GAMMELT TEMPEL.      SOLENS  
    LYS FALDER I EN STRIBE    NED GENNEM ET HUL I  
    LOFTET OG    RAMMER ET GADEFULDT RELIEF PA    DEN  
    FUGTIGE MOSTILGROEDE VÆG."
```

Denne tekst vil på skærmen (dersom skærbredden er 32) se sådan ud:

```
DU STAR I ET GAMMELT TEMPEL.  
SOLENS LYS FALDER I EN STRIBE  
NED GENNEM ET HUL I LOFTET OG  
RAMMER ET GADEFULDT RELIEF PA  
DEN FUGTIGE MOSTILGROEDE VÆG.
```

Altså har vi løst problemet – sådan da. For dels havde vi et pokers besvær med at sidde og tælle mellemrum for at snyde computeren til at »skifte linie« de rigtige steder, dels vil programmet (hvis det skrives direkte af) ikke kunne køres tilfredsstillende på andet end en computer med en skærbredde på 32. Hvis du fx har en ven med en computer med en skærbredde på 40, vil den på hans skærm komme til at se sådan ud:

```
DU STAR I ET GAMMELT TEMPEL.      SOLENS L  
YS FALDER I EN STRIBE    NED GENNEM ET HU  
L I LOFTET OG    RAMMER ET GADEFULDT RELI  
EF PA    DEN FUGTIGE MOSTILGROEDE VÆG.
```

Såre utilfredsstillende. Men vi *kan* slippe uden om problemet på en meget enkel måde: ved at *tage* lineskiftene med i vores tekst.

Det gør vi på følgende vis:

```
10 L$(20)="DU STAR I ET GAMMELT TEMPEL."+CHR$(13)+  
  "SOLENS LYS FALDER I EN STRIBE"+CHR$(13)+"NED  
  GENNEM ET HUL I LOFTET OG"+CHR$(13)+"RAMMER ET  
  GADEFULDT RELIEF PÅ"+CHR$(13)+"DEN FUGTIGE  
  MOSTILGROEDE VÆG."
```

På skærmen vil komme til at stå:

```
DU STAR I ET GAMMELT TEMPEL.  
SOLENS LYS FALDER I EN STRIBE  
NED GENNEM ET HUL I LOFTET OG  
RAMMER ET GADEFULDT RELIEF PÅ  
DEN FUGTIGE MOSTILGROEDE VÆG.
```

Så er den ved at være hjemme. Computeren kan nu – som følge af CHR\$(13), der er ASCII-koden for »vognretur«, se, hvor linierne skal deles, og den kan opbevare hele denne mundfuld i én variabel. Men der er et problem: hvis du forsøger at putte denne eller lignende variabler ind i et sæt ved hjælp af DIMensionering og DATA-sætninger, vil maskinen protestere vildt og voldsomt. Vi bliver nødt til at undlade at DIMensionere og i stedet skrive elementerne i sættet L\$() ud hver for sig:

```
10 L$(1)="AAAAA"+CHR$(13)+"BBBBB"  
20 L$(2)="CCCCC"+CHR$(13)+"DDDDD"
```

og så videre.

Dette vil maskinen acceptere, og du husker nok fra gennemgangen af demonstrationsspillet, at jeg sagde, at der var endnu en god grund til ikke at skrive lokalitetsbeskrivelserne ind i en

DATA-sætning – og her har du altså denne gode grund: nemlig, at dersom vi ønsker at gøre det hele nemmere for os selv ved at medtage lineskiftene i vores lokalitetsbeskrivelsesvariabler, bliver vi nødt til at gøre det på denne måde. Ulempen er, at vi optager en smule ekstra hukommelsesplads ved det, men fordelene er efter min mening så store, at de er det værd – også selv om der er endnu et irriterende minus ved dette system: hvis man prøver at gemme variabelsættet L\$() ved hjælp af en løkke, når man GEMmer SPillet, vil computeren (hvis veje ere uransagelige) kun gemme det, der står *foran* det første CHR\$(13). Men også dette er der en løsning på: vi lader simpelt hen *være* med at gemme lokalitetsbeskrivelserne, der som regel er ens hele spillet igennem.

Men hvad hvis der nu *er* ændringer i lokalitetsbeskrivelserne undervejs – ændringer, som stadig skal være der, når vi efter at have GEMt SPillet atter FORTsætter det? Hvis det hul, solens stråler falder ned igennem, bliver dækket til, og vi ønsker at ændre lokalitetsbeskrivelsen til »DU STÅR I ET GAMMELT TEMPEL. DU KAN IKKE SE EN PIND«. Hvad gør vi så?

Ja, så har vi jo fra forrige spil sektionen »ændringer«, hvor vi kan indføje den slags og sørge for, at det bliver læst igennem af computeren hele tiden, så den til stadighed ved, at solen er forsvundet, hvis visse betingelser er opfyldt. Og så betyder det ikke noget, at variabelsættet L\$() ikke bliver GEMt. Denne fremgangsmåde optager også en smule ekstra hukommelsesplads, men fordelene er stadig efter min mening så store, at det er det værd: Vi kan have alenlange lokalitetsbeskrivelser på flere linier, som vi kan jonglere rundt med som indekserede variabler. De er nemme at skrive (ikke noget med at tælle dumme mellemrum) – og systemet gør desuden, at vi kan have vores DATA for døre (verdenhjørner) stående *umiddelbart neden under* den lokalitetsbeskrivelse, de hører til. Du kan se dette system brugt i praksis i det næste spil i denne bog, »På skattejagt i Amazonlandet«.

En sidste lille ting angående skærmpresentation: Hvis du har en skærmbredde på så meget som 80 anslag, kan det være fristende at fylde hele skærmen ud med tekst i sin fulde bredde:

DU STAR I ET GAMMELT TEMPEL. SOLENS LYS FALDER I EN STRIBE NED GENNEM ET HUL I LOFTET OG RAMMER ET GADEFULDT RELIEF PÅ DEN FUGTIGE MOSTILGROEDE VÆG.

Fald ikke for denne fristelse! Det er nemlig sådan, at en skærmtekst er langt nemmere at læse, hvis linierne ikke er så lange. Det er ikke tilfældigt, at aviserne bruger spalter! Brug korte linier (ikke over 40 anslag lange):

DU STAR I ET GAMMELT TEMPEL.
SOLENS LYS FALDER I EN STRIBE
NED GENNEM ET HUL I LOFTET OG
RAMMER ET GADEFULDT RELIEF PÅ
DEN FUGTIGE MOSTILGROEDE VÆG.

Tidsfrist

Man kan forhøje spændingsmomentet i et spil væsentligt ved at indføre en tidsfrist. I »På skattejagt i Amazonlandet« er der fx en frist på ti dage, inden for hvilke man skal finde en skat. Hvis de ti dage er gået, er spillet forbi, og man er nødt til at begynde forfra. Det kan være temmelig nervepirrende kun at have en halv dag tilbage, hvis man er lige ved at løse det endelige problem – det er noget, der kan fremkalde det af enhver inkarneret spiller så eftertragtede adrenalinsus. Det er i princippet ret nemt at gøre: man indfører en variabel, der vokser med 1, hver gang det bliver midnat, og når den er oppe på 10, ved maskinen, at der er gået ti dage, hvorefter den afbryder spillet.

Her skal jeg lige skynde mig at nævne, at det, jeg snakker om, ikke er, hvad englænderne kalder *real time*, dvs. *rigtig* tid (selv om man ved hjælp af computerens ur også kan bruge dette som spændingsmoment), men *spilletid*.

Spilletiden styrer spillets forfatter selv ved at bestemme, at der for hver bevægelse går et vist antal minutter. Hvis du går fra lokalitet nr. 1 til lokalitet nr. 2, kan forfatteren bestemme, at der skal gå 10 minutter, inden du når frem. Det betyder ikke, at du



skal sidde i ti minutter og glo på en tom skærm, før der sker noget, men at der *i historien* er gået ti minutter. Lad os tage et illustrerende eksempel:

DATO: 1. JULI * KLOKKEN ER 12:00

DU STAR PÅ EN STRAND.

DU KAN GÅ:

NORDPÅ

HVAD VIL DU GØRE?

Gå nordpå

(Skærmen ryddes)

DATO: 1. JULI * KLOKKEN ER 12:10

DU STAR I EN SKOV.

DU KAN GÅ:

SYDPA

HVAD VIL DU GØRE?

Som du ser, er der på »uret« gået ti minutter, selv om du måske selv kun har brugt et halvt sekund på at skifte fra lokalitet til lokalitet. Det vil heller ikke tage dig lang tid at læse denne sætning: »Peter forlod stranden og stod ti minutter senere i skoven«, men for Peter er der gået ti minutter. Og hvis Peter slæber på noget tungt, vil der måske gå *tolv* minutter, før han når op i skoven. Kan vi også overføre *dette* til vores spil?

Javist, ja – mere om dette senere. Lad os først koncentrere os om selve uret.

Hvordan fortæller vi maskinen, hvor lang tid, der skal gå mellem de enkelte steder? Det gør vi på samme måde, som vi fortæller den, hvor man kommer hen, hvis man går i en bestemt retning. Lad os lige rekapitulere:

```
10 DIM L$(2),L(2,4)
20 FOR I=1 TO 2:FOR J=1 TO 4:READ L(I,J):NEXT J:
   NEXT I
30 L$(1)="PÅ EN STRAND. "
40 DATA 2,0,0,0
50 L$(2)="I EN SKOV. "
60 DATA 0,1,0,0
```

Dette lille program fortæller, hvad vi lige har set: at vi fra Lokalitet nr. 1, som er en strand, kan gå nordpå til Lokalitet nr. 2, som er en skov – og herfra kan vi gå sydpå tilbage til stranden.

Lad os prøve at lave en smule om på det:

```

10 DIM L$(2),L(2,8)
20 FOR I=1 TO 2:FOR J=1 TO 8:READ L(I,J):NEXT J:
    NEXT I
30 L$(1)="PÅ EN STRAND."
40 DATA 2,0,0,0,10,0,0,0
50 L$(2)="I EN SKOV."
60 DATA 0,1,0,0,0,10,0,0

```

Vi har udvidet sættet L() til at omfatte 2 gange 8 elementer, hvor det før kun bestod af 2 gange 4 elementer. De sidste fire tal i hver DATAsætning er *tidsangivelser*, der viser, hvor mange minutter, det tager at gå i en bestemt retning. Hvis vi står på Lokalitet nr. 1 og vil gå nordpå, kigger maskinen i den DATA-sætning, som hører til Lokalitet nr. 1, på det første tal (som repræsenterer nord). Det er et 2-tal, og maskinen flytter os derpå (hvis programmet vel at mærke har en bevæge-rutine med – hvilket dette ikke har) til Lokalitet nr. 2. Samtidig lægger den (hvis den vel at mærke et andet sted i programmet får besked på det) 4 til 1 (positionen for nord) og kigger efter, hvad der står på femte position. Der står et 10-tal, og dette tital lægger den til minutterne, således at der på skærmens »ur« er gået ti minutter. Står vi på Lokalitet nr. 2 og beder om at gå mod syd, kigger maskinen på den anden DATA-sætnings position nr. 2 (der repræsenterer syd), registrerer et 1-tal og flytter os til Lokalitet nr. 1, aflæser position nr. 2+4 (altså position nr. 6), hvor der står et 10-tal – og lægger igen 10 til minutterne. Hvis vi går fra stranden til skoven og tilbage igen, går der altså 2 gange 10 minutter. Ligesom vi bør kunne gå tilbage mod nord, hvis vi er gået mod syd, bør vi også bruge den samme tid på det begge veje. Også denne fremgangsmåde kan du studere nærmere i det næste spil, »På skattejagt i Amazonlandet«.

Nu ved vi altså, hvordan vi kan få maskinen til at »tælle minutter« og skabe en illusion af, at der går et stykke tid. Men hvordan får vi den til at skrive det ud på skærmen, så vi kan følge med i, hvor lang (spille)tid, der er gået?

Lad os kigge på et nyt program, der ordner dette (for det er ikke helt så ligetil, som du måske vil tro):

```
10 REM UR *****
20 MI=720:DA=1
30 TM=INT(MI/60)
40 MN=INT(MI)-(TM*60)
50 IF MI>1439 THEN 70
60 GOTO 80
70 MI=MI-1440:TM=TM-24:DA=DA+1
80 MN$=STR$(MN)
90 TM$=STR$(TM)
100 DA$=STR$(DA)
110 PRINT "DATO: ";DA$; ". JULI * ";
120 A$=RIGHT$(MN$,2)
130 IF LEN(MN$)=2 THEN A$="0"+RIGHT$(MN$,1)
140 PRINT "KLOKKEN ER";TM$; ":";A$
```

20: Vi sætter datoen til 1. juli klokken 12. 12 timer er 720 minutter, så vi sætter minutterne (variablen MI) til 720 og datoen (variablen DA) til 1.

30-40: Vi skal imidlertid også operere med timer, og eftersom der går 60 minutter på en time, finder vi timetallet ved at dividere MI med 60 og runde ned til nærmeste heltal (hvis MI fx var 725, ville MI/60 blive 12,083333 – og INT(MI/60) blive 12 ... og det er 12-tallet, vi er interesseret i.

50-70: Hvis MI er større end 1439, er der gået 24 timer, og vi skal have skiftet dato. Det sker i linie 70, hvor vi trækker 1440 fra MI (vi begynder jo på en ny dag og skal til at tælle minutterne forfra), trækker 24 fra TM (timeantallet) og lægger 1 til datoen.

80-100: Når computeren PRINTER tal ud på skærmen, laver den

altid et tomt anslag foran dem. Da vi ikke er interesseret i, at der kommer til at stå »KLOKKEN ER 12: 0«, bliver vi nødt til at forvandle tallene til strenge, så vi kan pille ved dem.

110: Vi PRINTer datoen ud. På skærmen vil komme til at stå:
DATO: 1. JULI *

120-130: Selv om vi har lavet tallene om til strenge, vil der stadig komme til at stå »KLOKKEN ER 12: 0« eller (hvis minutantallet er fx 13) »KLOKKEN ER 12: 13« – fordi maskinen har taget tomrummet foran tallet med i strengen. Vi fjerner i linie 120 det dumme hul mellem kolonnet og tallet ved at gøre A\$ lig med de to sidste tegn i strengen. Det duer ikke, hvis tallet er under 10, og det tager vi højde for i linie 130, hvor vi også sætter det manglende nul foran de etcifrede tal.

140: Computeren PRINTer på skærmen: KLOKKEN ER 12:00

Også dette ur kan du se i praksis i det næste spil, »På skattejagt i Amazonlandet«.

Egenskaber

I et brætrollespil som »Dungeons and Dragons« har hver spiller forskellige »egenskaber« eller »aktiver«, som inden spillet afgøres ved terningekast. Det kan dreje sig om fx. styrke, udholdenhed, formue, behændighed eller held. Hvis man tager en skala fra 1 til 10, vil en person med en styrke på 5 være svagere end en person med en styrke på 8 – det siger sig selv – ligesom en person med 10 guldmønter vil være rigere end en med 3 guldmønter.

Kan vi bruge dette i et eventyrspil? Ja, selvfølgelig. En måde, vi kan bestemme fx personens styrke på, er ved at lade tilfældighedstalsgeneratoren udpege et tal mellem 1 og 10. Det samme kan man så gøre med andre egenskaber såsom behændighed, held, osv. – hvorefter man i spillet lader disse tal afgøre, hvor-

dan det vil gå en person, når han møder fx. en varulv. Hvis han har en styrke på 1, en behændighed på 2 og et held på 3, er der en overvejende sandsynlighed for, at varulven vinder kampen – men hvis han har en styrke på 10, en behændighed på 10 og et held på 10, vil spilleren efter al sandsynlighed vinde. Det eneste, vi skal gøre, er at lade maskinen sammenligne modstanderens egenskabs-tal med spillerens og lade den stærkeste af dem vinde.

Vi kan også fra begyndelsen *selv* fastsætte visse tal, der én gang for alle afstikker hovedpersonens egenskaber. I »På skattejagt i Amazonlandet« har jeg fx givet min person en *bæreevne* på 8, og det betyder i praksis, at dersom han bærer ting, hvis samlede vægtværdi er over 7, vil han ikke kunne bære på mere. Det har en stor praktisk betydning, hvis man vil lave et spil, der er nogenlunde realistisk (så helten ikke kommer til at slæbe rundt på femten forskellige våben, tre skjolde, en 50-kilos melsæk, en robåd og en skabskuffert) eller blot gøre det sværere for spilleren og tvinge ham til at træffe et valg, når han får at vide, at han bliver nødt til at lægge en af de ting, han bærer på – en ting, han meget vel kan få brug for senere. Jeg har også i spillet medtaget en slags styrketal. Når spillet begynder, er spilleren frisk og udhvilet og kan bære på mere, end når han er ved at segne af træthed eller ikke har fået noget at spise i flere dage – og på samme måde kan han få flere kræfter ved at sove eller spise – ja, faktisk *tvinger* jeg ham til at sove otte timer en gang i døgnet (for at gøre spillet så realistisk som muligt), men maden må han selv finde ud af at skaffe sig. Det er et af de problemer, der kan gøre livet virkelig surt for ham. Tro mig.

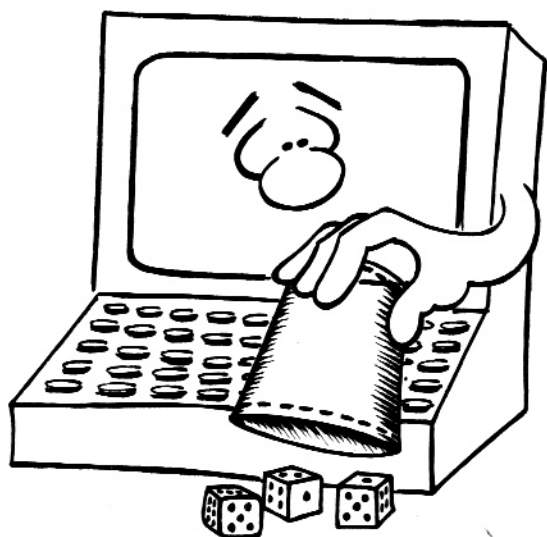
Mere om dette senere i kommentarerne til selve spillet. Hvis du blot har forstået grundprincipperne, vil du (ved at skele til mine eksempler i Amazonspillet) også kunne finde ud af at bruge dem, når du laver dine egne eventyr.

Tilfældigheder

Jeg nævnte lige før tilfældighedstalsgeneratoren, og den kan vi få megen nytte af. Ved hjælp af den kan vi lade begivenheder

indtræffe, uden at spilleren kan forudsige hvornår (selv om han har spillet spillet før), eller vi kan anbringe tingene i spillet tilfældigt, så de ikke ligger samme sted, næste gang han spiller det. Men lad aldrig maskinen flytte døde ting rundt *under* et spil. Det er spillerens opgave, og alt andet ville være snyd! I »På skattejagt i Amazonlandet« bliver en del af tingene (faktisk alle dem, der ikke *behøver* ligge på et bestemt sted) tilfældigt anbragt i begyndelsen af spillet – og dér bliver de liggende indtil næste spil (og hvis du GEMmer SPillet og henter det frem igen senere og FORTsætter det, vil disse ting stadig ligge de samme steder). Men spillet indeholder også en række levende væsener – fx en leopard – og dem kan man naturligvis lade bevæge sig rundt, så tosset man vil.

Hvordan gør man så det? Jo, hvis vi kalder den variabel, der bestemmer hvor fx leoparden skal opholde sig, for LE og giver maskinen ordre til hele tiden at give LE nye værdier ved hjælp



af tilfældighedstalsgeneratoren, så vil vi kunne skabe en illusion om, at leoparden bevæger sig uafhængigt rundt i spillelandskabet. Vi har så at sige skabt en »modspiller«. Hvis maskinen bestemmer, at LE er lig 8, vil leoparden i spillet blive flyttet til Lo-

kalitet nr. 8 – og hvis spilleren samtidig befinder sig på Lokalitet nr. 8, vil han blive overfaldet af leoparden og må så finde ud af, hvordan han uskadeliggør den. Man kunne fx lade ham skyde efter den og så lade tilfældighedstalsgeneratoren bestemme, om han rammer eller ej.

Mange computere sørger selv for at styre deres tilfældighedstalsgenerator ved at hente det nødvendige grundtal i maskinens ur, men nogle (fx. dem der kører på Microsoft BASIC), kræver i begyndelsen af spillet et »Random Number Seed« (»Tilfældighedstals-frø«), et grundtal, som skal bruges ved beregningen af de tilfældige tal i resten af spillet. Så helt tilfældigt er det altså i virkeligheden ikke.

Hvis du rynker på panden og ikke aner, hvad det er, jeg snakker om, er det nok fordi din maskine *selv* plejer at beregne grundtallet ud fra sit indbyggede ur, og så kan du bare hoppe videre til næste afsnit.

Men hvis du *har* en maskine, der kræver dette grundtal, og hvis du synes, at det er irriterende at skulle stikke maskinen et nyt hele tiden (hver gang du påbegynder et nyt spil), eller hvis du ikke vil friste nogen (eller måske dig selv) til at snyde, hvis de har fundet et »nemte« grundtal, som de kan bruge igen og igen, kan du bruge følgende fremgangsmåde:

```
10 T=(VAL(RIGHT$(TIME$,2))+1)*(VAL(MID$(TIME$,4,2))
    *(VAL(LEFT$(DATE$,2)))
20 RANDOMIZE T
```

Vi tager sekunderne, ganger dem med minutterne og ganger det tal, vi får, med dagen, hvorefter vi RANDOMIZERer det tal, vi derefter får (T). Derved får vi et grundtal, der er baseret på uret, og som næppe vil blive det samme to gange i træk. Og maskinen vil være tilfreds og ikke spørge os om noget grundtal.

Hvis du skulle være så heldig at have fx en DOS 2.0 version af Microsoft BASIC, kan du nøjes med at benytte dig af kommandoen RANDOMIZE TIMER, der uden videre omsvøb henter grundtallet i maskinens indbyggede ur. Check din håndbog!

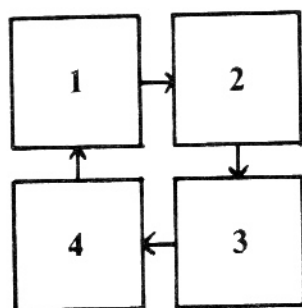
Dersom din maskine kører på en mindre avanceret BASIC, der mangler funktionerne TIMER eller TIME\$ og DATE\$, som giver adgang til ur og dato, skal du ikke fortvivle. Der findes nemlig en anden måde, vi kan få et rimelig tilfældigt grundtal på:

```
10 PRINT "INDLEDNING."  
20 PRINT:PRINT "TRYK PÅ MELLEMRUMSTASTEN,":PRINT  
   "NÅR DU VIL FORTSÆTTE."  
30 FOR I=1 TO 30000  
40 A$=INKEY$:IF A$="" THEN NEXT I  
50 RANDOMIZE I  
60 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT  
   CHR$(147)  
70 PRINT "INDLEDNINGEN FORTSÆTTER."
```

Når jeg i linie 10 har skrevet »INDLEDNING«, er det blot for at vise, at her skriver du indledningen til dit spil, og den må gerne fylde en hel skærm og være meget indviklet, så det tager spilleren tid at læse den. Når han har læst færdig, vil han blive mødt af ordene i linie 20 (eller 120 eller 150 – alt efter hvor mange linier din indledning fylder): TRYK PÅ MELLEMRUMSTASTEN, NÅR DU VIL FORTSÆTTE. Men mens han har siddet og læst, er maskinen for længst nået til linie 30 og 40, hvor den er begyndt at tælle løs fra 1, hvilket den vil blive ved med, indtil den når 30000. Det tager et pænt stykke tid, og den vil rimeligvis blive afbrudt af spilleren, inden den bliver færdig. Og i det øjeblik han trykker på mellemrumstangenten, hopper den ud af løkken, og det I, den er nået til (det kunne fx være 211) vil så blive brugt som grundtal for tilfældighedsberegninger, efter at det er blevet RANDOMIZERet i linie 50. Det går så hurtigt, at spilleren ikke vil bemærke andet, end at skærmen ryddes, når han trykker på mellemrumstasten, hvorefter INDLEDNINGEN FORTSÆTTER. Også med denne fremgangsmåde vil grundtallet næsten med garanti blive et nyt hver gang, for maskinen tæller *meget* hurtigt.

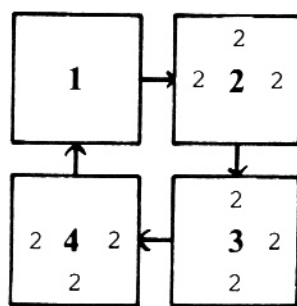
Labyrinter

Labyrinter er altid spændende, og der kommer et langt mere udtømmende kapitel om dem senere (Kapitel 6), men lad os her lige kigge på en »snyde-labyrint«, som er meget nem at lave, men som det ikke desto mindre kan være næsten umuligt at finde ud af igen. Lad os tage et kort med fire rum:



Som du kan se, har jeg lavet »dørene« ensrettede, så der kun er én måde at gå gennem rummene på: fra Rum nr. 1 til Rum nr. 2 til Rum nr. 3 til Rum nr. 4 og tilbage til Rum nr. 1. Hvis vi siger, at Rummene 2, 3 og 4 skal være labyrinten, kan man altså kun komme ud af den (når man er gået ind i den fra Rum nr. 1 og står i Rum nr. 2) ved at gå SYDPÅ, VESTPÅ og NORDPÅ. Og hvis vi er så nederdrægtige at narre spilleren til at tro, at han kan gå i alle fire verdenshjørner i stedet for kun ét, vil han for alvor fare vild. Men hvor skal vi så anbringe ham, hvis han står i Rum nr. 3 og skriver GÅ ØSTPÅ – for han *kan* jo ikke gå østpå fra Rum nr. 3?

Jo, vi anbringer ham i Rum nr. 2, hver gang han vælger en forkert retning. Lad os vise dette på et andet kort:



Her viser de små totalter, hvor man havner (i Rum nr. 2), hvis man ikke tager en »reglementeret« retning.

Men vil man ikke på lokalitetsbeskrivelsen kunne se, hvor man er og lynhurtigt kunne finde ud af at orientere sig? Nej, ikke hvis vi gør lokalitetsbeskrivelserne for de tre labyrintrum (2, 3 og 4) ens, fx: DU STÅR I EN FORVIRRENDE LABYRINT AF KLIPPEGANGE. Så vil spilleren ikke kunne se, at han i virkeligheden er gået tilbage mod nord fra Rum nr. 3, hvis han har bedt om at GÅ ØSTPÅ. Det fungerer aldeles tilfredsstillende i praksis – tro mig. Her ser du, hvordan vi skriver en sådan »snyde-labyrint« ind i et program:

```

10 L$(1)="STAR UDEN FOR EN LABYRINT."
20 DATA 0,0,2,0
30 L$(2)="STAR I EN FORVIRRENDE LABYRINT AF
   KLIPPEGANGE."
40 DATA 2,3,2,2
50 L$(3)=L$(2)
60 DATA 2,2,2,4
70 L$(4)=L$(2)
80 DATA 2,2,1,2

```

DATA-sætningerne er naturligvis for verdenshjørner (se gennemgangen af demonstrationsspillet), og maskinen vil, hver gang spilleren står i enten Rum nr. 2, 3 eller 4 skrive

DU KAN GÅ:

NORDPA

SYDPA

ØSTPA

VESTPA

– og hvis spilleren vælger et forkert af disse verdenshjørner, vil han blive transporteret tilbage til Rum nr. 2 uden at ane noget om det. Hvis du tror, det er en let sag at finde frem til den rigtige kombination af verdenshjørner (i dette tilfælde S-V-N) og slippe ud af labyrinten, kan du godt tro om igen. Det er faktisk så svært, at du et eller andet sted i spillet bør lægge et fingerpeg til spilleren om, hvordan han skal klare labyrinten – selv om der findes en måde at gøre det på: Man kan lægge nogle af de ting, man går med, i hvert rum. Så vil man hurtigt finde ud af labyrintens konstruktion. Men hvis nu labyrinten er på 10 rum, og man kun har to ting på sig . . . ! Jeg synes stadig, det vil være mest fair at lægge et fingerpeg til spilleren et eller andet sted – men du kan naturligvis gøre, hvad du vil.

Vi vender tilbage til »rigtige« labyrinter i kapitel 6, men en af denne slags (en »snyde-labyrint«) kan du også se demonstreret i »På skattejagt i Amazonlandet«, som vi vil gå over til i næste kapitel.

Anagrammer

Ordet »anagram« er af græsk oprindelse og bruges om et ord, der er dannet af et andet, ved at man bytter bogstaverne om. Således er TAK fx et anagram af KAT, og det samme er AKT. Faktisk kan bogstaverne K-A-T kombineres på 6 måder:

KAT

KTA

ATK

AKT

TKA

TAK

Hvis et ord består af 4 bogstaver, kan de kombineres på 4 gange så mange måder som 3 bogstaver, altså 4 gange $6 = 24$ måder. Hvis et ord består af 5 bogstaver, kan de kombineres på 5 gange så mange måder som 4 bogstaver, altså 5 gange $24 = 120$ måder – og så fremdeles.

Men hvad i alverden kommer det os ved?

Det kommer os ikke specielt ved . . . men der er bare det gode ved anagrammer, at computere har nemt ved at lave dem – og at vi kan have svært ved at hitte rede i dem. Vi kan med andre ord bruge dem som opgaver eller måske fingerpeg i vores eventyrspil. I spillet »Faraos skat«, som følger senere i denne bog, har jeg fx en mumie, der forlanger, at man siger dens navn, og som et fingerpeg giver den de tre bogstaver, navnet består af, i forkert rækkefølge. Man kunne også bruge tal-anagrammer i et spil. Lad os sige, at spilleren skal demontere en bombe, inden der er gået to minutter (her kunne man i øvrigt bruge *real time* som en realistisk effekt – måske ligefrem vist som et ur, der sekund for sekund tæller ned øverst på skærmen). Han får . . . lad os sige 4 tal, som han skal sætte sammen til den rigtige kombination, og eftersom der er 24 muligheder, får han travlt. Man kunne også bruge tal-anagrammer som fingerpeg til en pengeskabskombination . . . mulighederne er mange. Her er et lille program, der laver ord-anagrammer på tre bogstaver:

```
10 RANDOMIZE (VAL(RIGHT$(TIME$,2))+1)*  
    (VAL(MID$(TIME$,4,2))*VAL(LEFT$(DATE$,2)))  
20 FOR I=1 TO 3:A(I)=INT(RND*26+65)  
30 FOR J=1 TO I-1  
40 IF A(I)=A(J) THEN A(I)=INT(RND*26+65):GOTO 30  
50 NEXT J  
60 FOR K=1 TO 3  
70 IF A(K)=65 OR A(K)=69 OR A(K)=73 OR A(K)=79 OR  
    A(K)=85 OR A(K)=89 THEN OM=1
```

```

80 IF OM=0 THEN 20
90 NEXT K
100 A$=CHR$(A(I))+A$
110 NEXT I
120 FOR I=1 TO 3:B$(I)=MID$(A$,INT(RND*3+1),1)
130 FOR J=1 TO I-1
140 IF B$(I)=B$(J) THEN
    B$(I)=MID$(A$,INT(RND*3+1),1):GOTO 130
150 NEXT J:NEXT I
160 FOR I=1 TO 3
170 C$=B$(I)+C$
180 NEXT
190 IF LEN(C$)>3 THEN C$="":GOTO 120
200 IF C$=A$ THEN 120
210 PRINT C$;" ER ET ANAGRAM AF:"
220 INPUT D$
230 IF D$=A$ THEN PRINT "RIGTIGT.":END
240 PRINT "FORKERT. OM IGEN!":PRINT:GOTO 210

```

Linie for linie:

10: Her beregnes et grundtal til tilfældighedstalsgeneratoren ved hjælp af ur og dato. Hvis du har en Commodore eller en lignende datamat, der selv beregner grundtallet, skal du springe denne linie over. Hvis du har en BASIC-version, der ikke har funktionerne TIME\$ og DATE\$, så se i afsnittet »Tilfældigheder« tidligere i dette kapitel, hvordan man kan frembringe dette grundtal på anden vis.

20: Her udtrækkes tilfældigt tre tal mellem 65 og 90. 65 er ASCII-koden for »A« og 90 er ASCII-koden for »Z«.

30-50: Her sørges for, at de tre tal bliver forskellige.

60-90: Her sørges for, at mindst et af de tre tal kommer til at svare til ASCII-koden for en vokal (Æ, Ø og Å dog undtaget, da

mange computere ikke har disse vokaler). Du kan slå op i en tabel over ASCII-koderne og se, at 65 svarer til »A«, 69 til »E«, 73 til »I«, 79 til »O« og 85 til »U«. Er der mindst én vokal med i et ord, er der en stor chance for, at det kommer til at lyde »sand-synligt«.

100-110: Her omdannes de tre tal til bogstaver, lægges sammen til et ord (A\$), og I-løkken afsluttes.

120-180: Her dannes anagrammet C\$ af det ord, der blev dannet i linie 100.

190: En lille sikkerhedsforanstaltning. Hvis C\$ skulle vise sig at blive på mere end tre bogstaver (det er faktisk på besynderlig vis hændt et par gange på min maskine), så om igen!

200: Hvis C\$ er lig med A\$, har vi ikke noget anagram. Om igen!

210-240: Her kan vi personligt prøve at løse anagram-gåden.

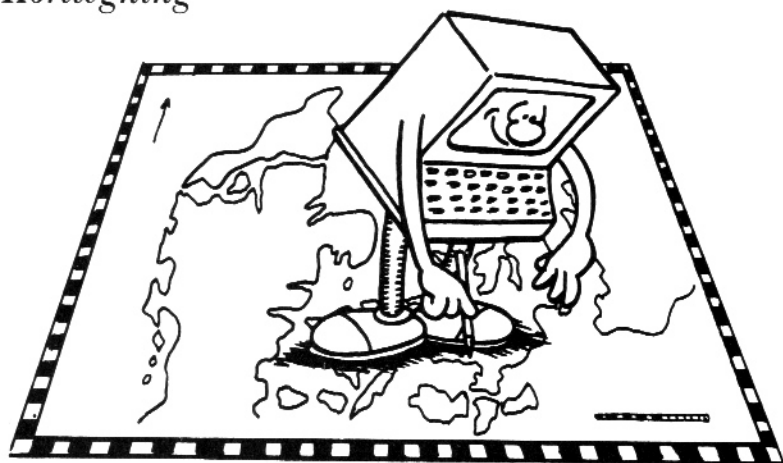
Store/små bogstaver

Visse computere (fx. IBM PC) skelner mellem store og små bogstaver, og det kan være irriterende at få en afvisende meddelelse, når man skriver »gå sydpå«, fordi maskinen kun vil acceptere »GÅ SYDPÅ«. For at skære dette ud i pap for spilleren kan man i begyndelsen af sine spil lægge følgende lille program:

```
10 INPUT A$:REM KAN FX VÆRE DET INPUT, HVOR DU BEDER  
   OM SPILLERENS NAVN  
20 IF A$="" THEN 10:REM FOR AT IMØDEGÅ, AT DER BLOT  
   SVARES MED ET TRYK PÅ RETURTASTEN  
30 FOR I=97 TO 125:FOR J=1 TO LEN(A$)  
40 IF ASC(MID$(A$,J,1))=I THEN PRINT "STORE  
   bogstaver, TAK!":GOTO 10
```

OBS! Dette program er skrevet til en maskine, hvor bogstaverne æ, ø og å har ASCII-værdierne 123, 124 og 125. Hvis du ikke har disse bogstaver på din maskine eller hvis de har andre ASCII-værdier, skal du lave løkken FOR I=97 TO 125 i linie 30 om til FOR I=97 TO 122 og bagefter eventuelt lave en tilføjelse, der drager de ASCII-værdier, æ, ø og å har på din maskine (hvis du altså har dem), med i beregningen.

Korttegning



Som sidste finesse går vi over til korttegningens ædle kunst. Når en spiller går i gang med et eventyrspil med mange lokaliteter, vil han hurtigt opdage, at det er nødvendigt at tegne et kort for at kunne finde rundt i spillelandskabet. Somme vil hævde, at dette er en del af charmen ved eventyrspil – det, at man ligesom en ægte opdagelsesrejsende kan kortlægge nyt og ukendt land – hvorimod andre finder det dødirriterende at skulle sidde med papir og blyant, når det nu er på *skærmen*, det foregår. Til benefit for denne sidste kategori af spillere kan man (hvis man har plads i computerens hukommelse), indlægge et kort, så spilleren

slipper for at fjerne fingrene og opmærksomheden fra tasterne.

Nu svor jeg ganske vist højt og helligt i indledningen, at jeg ikke ville bruge indviklede grafik-kommandoer i denne bog, og det vil jeg sådan set heller ikke. Til tegning af kortet benytter vi os af semigrafik – vi lader computeren PRINTe tegningen, fuldstændig som vi lader den PRINTe bogstaver. De semigrafiske tegn, vi vil benytte, kan du se i nedenstående udskrift.

Men før vi begynder at konstruere kortet, må vi lige fastlægge en gylden grundregel:

Kortet må ikke på noget tidspunkt vise flere lokaliteter end dem, spilleren har udforsket.

Dette siger sådan set sig selv. Hvis spilleren ved spillets begyndelse blot behøver at trykke på en tast for at se *hele* kortet, er der jo ikke noget at udforske! For at lette forståelsen (og indtastningen), har jeg konstrueret et appendiks til »Huset med de ti rum«, en programudvidelse, som tegner et kort over de udforskede lokaliteter. Hvis du blader tilbage til side 18 og studerer den nydelige tegning, vil du kunne se, hvordan dette kort skal se ud (eller kig på skærmudskriften sidst i dette afsnit).

Lad os nu først LISTe dette tillægsprogram (der vel at mærke *ikke* kan køre alene, men *skal* skrives sammen med »Huset med de ti rum«!) og så bagefter gennemgå det linie for linie:

```
10 REM KORTTEGNINGSPROGRAM - APPENDIKS TIL "HUSET  
MED DE TI RUM". KAN IKKE KØRE SELVSTÆNDIGT.  
445 GOSUB 1100  
555 IF B$="K" THEN GOTO 1400  
925 K(L0)=1  
1099 REM DATA FOR KORT *****  
1100 DIM K(AL), K$(AL,5)  
1105 REM COMMODORE-BRUGER, SKRIV HER: DIM X$(69)  
*****  
1110 FOR I=1 TO AL:FOR J=1 TO 5:READ K$(I,J):NEXT J:  
NEXT I
```

1120 DATA " _____ "
 1121 DATA " |RUM | "
 1122 DATA " |NR.1 | "
 1123 DATA " | | "
 1124 DATA " _____ "
 1130 DATA " _____ "
 1131 DATA " |RUM | "
 1132 DATA " |NR.2 | "
 1133 DATA " | | "
 1134 DATA " _____ "
 1140 DATA " _____ "
 1141 DATA " |RUM | "
 1142 DATA " |NR.3 | "
 1143 DATA " | | "
 1144 DATA " _____ "
 1150 DATA " _____ "
 1151 DATA " |RUM | "
 1152 DATA " |NR.4 | "
 1153 DATA " | | "
 1154 DATA " _____ "
 1160 DATA " _____ "
 1161 DATA " |RUM | "
 1162 DATA " |NR.5 | "
 1163 DATA " | | "
 1164 DATA " _____ "
 1170 DATA " _____ "
 1171 DATA " |RUM | "
 1172 DATA " |NR.6 | "
 1173 DATA " | | "
 1174 DATA " _____ "
 1180 DATA " _____ "
 1181 DATA " |RUM | "
 1182 DATA " |NR.7 | "
 1183 DATA " | | "
 1184 DATA " _____ "

```

1190 DATA " _____ "
1191 DATA " |RUM | "
1192 DATA " |NR. 8| "
1193 DATA " | | "
1194 DATA " _____ "
1200 DATA " _____ "
1201 DATA " |RUM | "
1202 DATA " |NR. 9 | "
1203 DATA " | | "
1204 DATA " _____ "
1210 DATA " _____ "
1211 DATA " |RUM | "
1212 DATA " |NR.10| "
1213 DATA " | | "
1214 DATA " _____ "
1215 RETURN

1399 REM KORTTEGNINGSROUTINE *****
1400 CLS:REM COMMODORE-BRUGER, SKRIV: PRINT
      CHR$(147) *****
1401 RESTORE 1120:REM COMMODORE-BRUGER, SKRIV KUN:
      RESTORE *****
1402 REM COMMODORE-BRUGER, SKRIV HER: FOR I=1 TO
      69:READ X$(I):NEXT I *****
1403 FOR I=1 TO 10:FOR J=1 TO 5:READ K$(I,J):
      NEXT J:NEXT I
1404 FOR I=1 TO 10:FOR J=1 TO 5
1407 IF K(I)=0 THEN K$(I,J)="      "
1408 NEXT J:NEXT I
1420 FOR J=1 TO 5:FOR I=1 TO 4
1424 PRINT K$(I,J);
1430 IF I=4 THEN PRINT
1440 NEXT I:NEXT J
1500 FOR J=1 TO 5:FOR I=5 TO 8
1507 PRINT K$(I,J);
1510 IF I=8 THEN PRINT

```

```

1520 NEXT I:NEXT J
1550 FOR J=1 TO 5:FOR I=9 TO 10
1555 IF I=9 THEN PRINT TAB(8):REM COMMODORE-BRUGER,
      SKRIV I STEDET: IF I=9 THEN PRINT TAB(7) *****
1570 PRINT K$(I,J);
1580 IF I=10 THEN PRINT
1590 NEXT I:NEXT J
1702 PRINT:PRINT:GOSUB 2000:GOTO 460
2000 PRINT:PRINT:PRINT:PRINT
2005 PRINT "TRYK PÅ MELLEMRUMSTASTEN,"
2010 PRINT "NÅR DU ØNSKER AT FORTSÆTTE."
2015 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I
      STEDET: GET A$ *****
2020 IF A$="" THEN 2015
2030 IF A$=" " THEN RETURN
2040 GOTO 2015

```

Linie for linie:

445: Herfra hoppes ned til indlæsning af data for kort.

555: Hvis man, når man får spørgsmålet »Hvad vil du gøre?«, skriver »K«, aktiveres korttegningsrutinen.

925: Det indekserede variabelsæt K() kontrollerer, hvilke af kortets lokaliteter, der skal tegnes. Hvis K=0, skal lokaliteten ikke tegnes. Hvis K=1, skal den tegnes. I denne linie, der kommer til at ligge først i skærmtekstrutinen, tilskrives K(LO) værdien 1, hvilket betyder, at når man har »besøgt« en lokalitet, har maskinen grønt lys til at tegne den med på kortet.

1100: Her dimensionerer vi de to nye indekserede variabelsæt K() og K\$(). Som du husker, står AL for Antallet af Lokaliteter – i dette tilfælde 10. K\$() indeholder DATA for lokalitetens »udseende« (se udskriftens linier 1120-1214). Angående K(): se forklaringen til linie 925.

1105: Hvis du har en Commodore, er du nødt til at oprette et »fup-DATAsæt« for at få RESTORE-kommandoen i linie 1401 til at fungere (forklaring følger).

1110: Indlæsning af DATAsættet K\$().

1120-1214: DATA for de enkelte lokaliteters »udseende« på kortet. Det er her, du kan se, hvilke semigrafiske tegn, du skal benytte. I udskriften hænger tegnene ikke helt sammen, men det vil de gøre på skærmen.

1215: DATA-indlæsningen er slut, og vi RETURNerer til hovedprogrammet.

1400: Korttegningsrutinen indledes med en skærmrydning.

1401-1402: Datasættet K\$() RESTOREs – dvs. gøres klar til en ny indlæsning. Dette er nødvendigt, da linie 1407 »ødelægger« sættet for hver korttegnings-runde. Hvis du opererer med Microsoft BASIC, kan du anføre, fra hvilken linie, du ønsker den nye indlæsning foretaget, ved at føje et linienummer til RESTORE-kommandoen, hvorimod en Commodore 64 kun vil acceptere selve RESTORE-kommandoen uden efterfølgende linienummer. Har du en Commodore, og bruger du RESTORE-kommandoen, vil den nye indlæsning begynde ved programmets *første* DATAsætning (i dette program sættet L()), og *det* er vi absolut ikke interesseret i. Så derfor foretager vi (og med »vi« mener jeg i dette tilfælde *kun* Commodore-brugerne) en fup-indlæsning af programmets første 69 DATA. De bliver anbragt i sættet X\$(), som vi oprettede i linie 1105 – og som vi absolut ikke skal bruge til andet! Når denne indlæsning er foretaget, er vi kommet til det rigtige sted (DATA nr. 70 i linie 1120), og vi kan gå over til den nye indlæsning af vores kort-DATA.

1403: Indlæsning af kort-DATA.

1404-1408: Vi checker, hvilke lokaliteter, der har været besøgt.

Hvis de ikke har været besøgt (hvis $K()=0$), kommer tegningen af den tilhørende lokalitet ($K\$()$) til at bestå af »tomme« felter.

1420-1590: Her PRINTes kortet ud, og det sker i tre etaper, da der er tre rækker af lokaliteter under hinanden:

Første række: Rummene 1,2,3 og 4.

Anden række: Rummene 5,6,7 og 8.

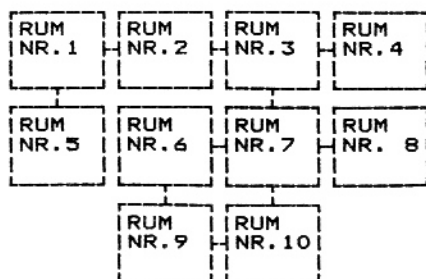
Tredje række: Rummene 9 og 10.

Bemærk, hvordan vi i linie 1555 får Rum nr. 9 flyttet hen på den rigtige plads (under Rum nr. 6) ved at bruge en TAB()-funktion.

Og vær venligst opmærksom på, at der *ikke* er nogen rum, der *ikke* bliver PRINTet ud. Det bliver de alle sammen, men dem, der ikke har været besøgt endnu, bliver PRINTet som tomme felter!

1702-2040: En rutine, der giver spilleren mulighed for at kigge på kortet, indtil han trykker på mellemrumstasten, hvorefter maskinen vender tilbage til hovedprogrammet.

En skærmudskrift af kortet over det fuldt udforskede hus vil se sådan ud:

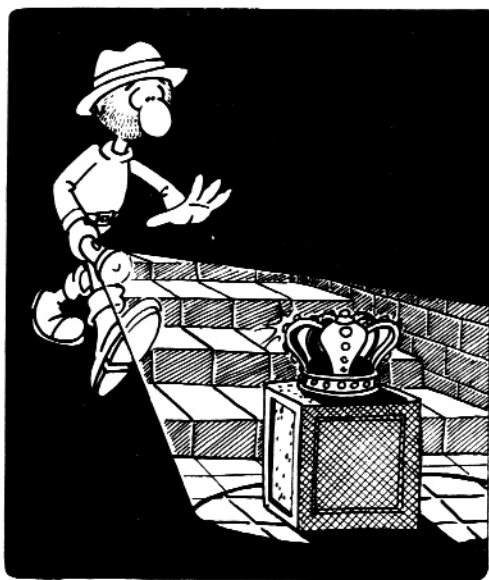


Bemærk, at kortet ikke vil få det »stiplede« udseende på selve skærmen. Der vil linierne hænge sammen, som de skal.

Hvis du har lavet et spil med et kort, der er så stort, at det ikke

kan være på skærmen på én gang, må du dele kortet op i to eller måske flere sektioner (dvs. faktisk operere med to eller flere kort) og nøjes med at lade maskinen PRINTe den sektion af kortet, som spilleren »befinder sig« på – muligvis med en indlagt besked til ham om, at dersom han vil se en anden del af kortet, skal han »bevæge sig« op/ned/hen til en af de lokaliteter, der er aftegnet på denne anden del af kortet.

5 · På skattejagt i Amazonlandet



Hvis du har sat dig grundigt ind i de indtil nu gennemgåede principper, burde du være i stand til at lave et super de luxe-eventyrspil med talrige krummelurer og finesser. Naturligvis sætter maskinens hukommelseskapacitet en grænse for, *hvor* langt og indviklet et spil kan blive (medmindre du har mindst én diskettestation, for så er mulighederne nærmest ubegrænsede), men selv på maskiner med en lille hukommelseskapacitet er der mulighed for at lave gode spil. Den virkelige grænse sættes nok snarere af forfatterens fantasi – afhænger af, om han (eller hun) er i stand til at udtænke en handling, der frembyder tilstrækkelig mange udfordringer for spilleren. Som tidligere nævnt kan jeg ikke gøre noget for at forbedre din fantasi, men hvad jeg kan gøre, er at vise dig, hvordan de principper, jeg har gennemgået,

kan bruges i praksis. Til dette formål har jeg konstrueret det næste spil, vi skal kigge på, »På skattejagt i Amazonlandet.«

Jeg vil nu give en synopsis af, hvad der sker i spillet, og hvis du ikke er interesseret i at få røbet handlingen, så er det nu, du skal hoppe frem til LISTningen af spillet. Men jeg kan ikke love dig, at du ikke finder ud af en hel del om spillet blot ved at skrive det af. Vi er endnu ikke færdig med lektionerne, og selv om der sidst i bogen er et detektivspil, »Mordet i den lukkede boks«, i hvilket jeg har kodet de sætninger, der kunne røbe for meget om handlingen, har jeg valgt ikke at gøre det samme her af hensyn til de læsere, der gerne vil have en nærmere forklaring på, hvad det er, der foregår. Og denne forklaring begynder med en synopsis, derefter kommer et kort over spillelandskabet, derefter LISTningen af spillet og til sidst en forklaring til de linier, der indeholder ting, som ikke er blevet uddybende gennemgået tidligere.

Synopsis: I »På skattejagt i Amazonlandet« skal spilleren finde en skat, en kostelig smaragdkrone, et sted midt ude i Amazonlandet. Han har ti dage (spilletid) til det, og hvis denne tidsfrist overskrides, vil han drukne, når en nyopført dæmning i området bliver åbnet. Spillet begynder med, at han bliver sat af en helikopter midt ude i urskoven uden noget som helst på sig. Det eneste, han i første omgang kan foretage sig, er at gå rundt i landskabet og udforske det. Tilfældigt anbragt ude i urskoven ligger bl.a. en riffel, og på en handelsstation kan vores spiller købe bl.a. ammunition (han *har* altså åbenbart nogle penge med), og først når han har både riffel og ammunition, vil han være udrustet til at skyde de vilddyr, der dukker op. Fx er der på Lokalitet nr. 6 en bjørn, som vogter en hule, man ikke kan komme ind i, før bjørnen er skudt, og der er en leopard, som tilfældigt kan dukke op overalt. Når man udsættes for en sådan trussel – leoparden, bjørnen og (til sidst i spillet) en panter – kan man skyde den eller flygte. Man har kun disse to muligheder i denne situation. Hvis man ikke har *både* riffel og ammunition, er man *nødt* til at flygte, og så vil man i en tredjedel af tilfældene blive indhentet og dræbt og i to tredjedele af tilfældene blive placeret på en lokalitet, der ligger *ved siden* af den, man flygtede fra. Man kommer altså ikke alt for langt væk, men til gengæld har man

smidt alt, hvad man bærer på, på det sted, man flygtede fra.

Fra tid til anden vil man få at vide, at det er længe siden, man har fået noget at spise, og jo længere tid, der går siden sidste måltid, jo mindre kan man bære på, og hvis der går mere end fire dage efter sidste måltid, dør man af sult. Så man er altså nødt til at skaffe sig mad undervejs. Det kan man gøre på to måder: man kan spise et af de dyr, man har skudt (og kun én gang, hvilket muligvis ikke er særlig logisk, da ingen kan sætte en hel leopard til livs på én gang, men til gengæld befordrende for spændingen, for så skal man til at lede efter mad igen) – men man skal naturligvis skyde dyret først, hvilket ikke er noget problem med bjørnen, der altid dør (den er i virkeligheden blot en hindring, der skal gøre det svært for en at komme ind i hulen), men leoparden er vanskelig at få ram på – i 3 ud af 4 tilfælde vil man ramme ved siden af, og leoparden vil flygte.

Og når man så har fået ram på et dyr, skal man have noget at partere det med – det kan gøres med en kniv, der er tilfældigt anbragt, eller en machete, der ligger inde i bjørnens hule. Er disse betingelser opfyldt, kan man spise dyret – og er det leoparden, man spiser, får man sig en overraskelse, men mere om dette senere.

Den anden måde, man kan få mad på, er ved at købe noget proviant på handelsstationen, men dér kan man desværre ikke få nogen dåseåbner. Der er udsolgt, og dåseåbneren ligger et andet sted (tilfældigt anbragt – selv *jeg* ved ikke hvor). Der er også den hage ved provianten, at man ikke kan bære den alene – det gælder i øvrigt også de døde dyr – så man finder hurtigt ud af, at det vil være en god idé at anskaffe sig nogle indfødte bærere – men hvor finder man dem? På handelsstationen er der ikke nogen, der har lyst til at bære ens ting, og det er der tilsyneladende heller ikke andre steder. Dog får man et sted den oplysning, at der er en tykning, der spærrer vejen mod vest, og hvis man UNDersøger TYKningen, vil man få at vide, at man kan hugge sig igennem den. Men med hvad? Hvis man har fundet kniven, finder man ud af, at den ikke kan bruges, men jeg fortalte lige før, at der ligger en machete i bjørnens hule, og det er *den*, man skal have fat i.

For at komme ind i bjørnens hule er man som sagt nødt til at skyde bjørnen først, og når det er sket, melder en ny forhindring sig: man kan ikke se noget derinde – i hvert fald ikke, hvis man ikke har den lommelygte, man også kan købe på handelsstationen. Hvis man har den, kan man se i mørke (men ikke, hvis man lægger den fra sig).

Godt. Vi har skudt bjørnen og træder nu med lommelygten i hånden ind i dens hule, og her løber vi ind i en labyrint – en »snyde-labyrint« af den type, jeg beskrev i forrige kapitel. Macheten ligger inde i labyrinten, og hvis man ikke har fået et fingerpeg om, hvordan man skal forcere den, kan man meget vel komme til at dø af sult derinde. Fingerpeget er i form af en dagbog, der ligger i nærheden af bjørnehulen, i en forladt lejr ved siden af skelettet af en opdagelsesrejsende, og i dagbogen har staklen engang skrevet, at han fór vild i bjørnens hule, men fandt ud igen, fordi han kom til at tænke på SØS. Siger det dig noget? Hvis ikke, kan jeg røbe, at det betyder syd-øst-syd, hvilket er koden for vejen gennem labyrinten.

Bevæbnet med macheten træder man fra bjørnens hule atter ud i junglen og kan nu hugge sig gennem tykningen, hvorefter man kommer til en landsby, hvor man kan leje nogle bærere.

I den opdagelsesrejsendes dagbog står også en anden nyttig oplysning: at han går med en guldnøgle om halsen, og at der er en stor leopard, der kredser om lejren. Skelettet har ikke nogen nøgle om halsen, og den logiske konklusion er naturligvis, at leoparden har ædt den opdagelsesrejsende og dermed nøglen, som den altså render rundt med i maven. Man er med andre ord nødt til at skyde leoparden for at få fat i nøglen. Guldnøglen skal bruges til at åbne et skrin (tilfældigt anbragt), som indeholder en sølvnøgle, der kan åbne en dør ind til den store hule, hvor skatten befinder sig.

Og hvis du tror, at spillet hermed er forbi, så kan du godt tro om igen – men inden jeg beskriver skattehulen, så lad mig lige fortælle endnu et par ting om junglen:

Hver gang, det bliver aften i spillet, bliver det naturligvis mørkt, og hvis man ikke har lommelygten med, er der 50 percents chance for, at der går en sky for månen, så man intet kan

se. Et sted ved en flodbred ligger der i øvrigt en båd, og man kan spare tid ved at sejle med den – men risikerer så til gengæld at sejle ud over et vandfald og blive slået ihjel.

En gang i døgnet (når klokken har passeret midnat) får man at vide, at man er træt og lægger sig til at sove, hvorefter spillet fortsætter otte (spille)timer senere. Det fremmer realismen, og så går der også dyrebar tid med det, så spilleren kommer endnu mere i tidnød.

På handelsstationen kan man også købe tændstikker, og dem skal man bruge til at antænde noget dynamit med senere i spillet. Dynamitten ligger tilfældigt anbragt et eller andet sted i spillet, og man kan ikke bære den alene – og eftersom man ikke kan finde skatten uden dynamitten, ja, så *kan* man altså ikke undvære bærerne og er altid nødt til at leje dem. Hvis man fyrer dynamitten af uden at lægge den fra sig, bliver man naturligvis slået ihjel – og hvis man lægger dynamitten fra sig og antænder den det forkerte sted, kan man lige så godt begynde forfra.

Når man er kommet ind i hulen, når man snart en afgrund og kan ikke komme længere. Men et andet sted er der en hængebro over en kløft, og her kan man tage et tov, der hænger ned fra broen, og ved hjælp af dette fire sig ned i afgrunden. Man kan i øvrigt kun finde tovet ved at UNDERsøge BROen, og hvis man prøver at bruge macheten på broen, styrter man ned og slår sig ihjel.

Men når man har firet sig ned i afgrunden, er man kommet et stykke nærmere på skatten (i øvrigt skal man herefter ikke prøve på at TAGe TOVet, for så ryger det ned, og man kan ikke slippe ud). Hernede i den nedre hule er der en masse gange at udforske, men på et eller andet tidspunkt kommer man til en tilmuret indgang, og bag denne ligger skatten. Det er her, man skal sprænge dynamitten.

Indgangen ryger i luften, og man træder med sine bærere ind i skatkammeret. Spillet er forbi – eller hvad?

Åh, nej . . . for det første, man møder, er en frygtindgydende totem, og da de overtroiske bærere ser den, flygter de rædselsslagne, ja, de smider faktisk (næsten) alting – inklusive lommelygten – fra sig. Når man har fundet lommelygten og kan se sig omkring, bemærker man skatten – en grøn smaragdkrone – og en

flaske med whisky. Derudover bemærker man også, at sølvnøglen (den til huledøren) er forsvundet. Den har de indfødte nemlig taget med sig, og i deres panik har de låst døren udefra.

Nu er gode råd dyre.

Hvis man havde UNDERsøgt SØLvnøglen, ville man have fået at vide, at der på den stod: »Der findes en kopi af mig.« Ens eneste chance for at slippe ud af hulen er altså at finde denne kopi af nøglen, men hvor er den?

Et helt andet sted i hulen er der et hul i en væg, og man skal tømme whiskyflasken ned i dette hul for at udløse en skjult mekanisme, der får et panel til at glide til side, og bag dette panel ligger en jernnøgle, der passer til huledøren.

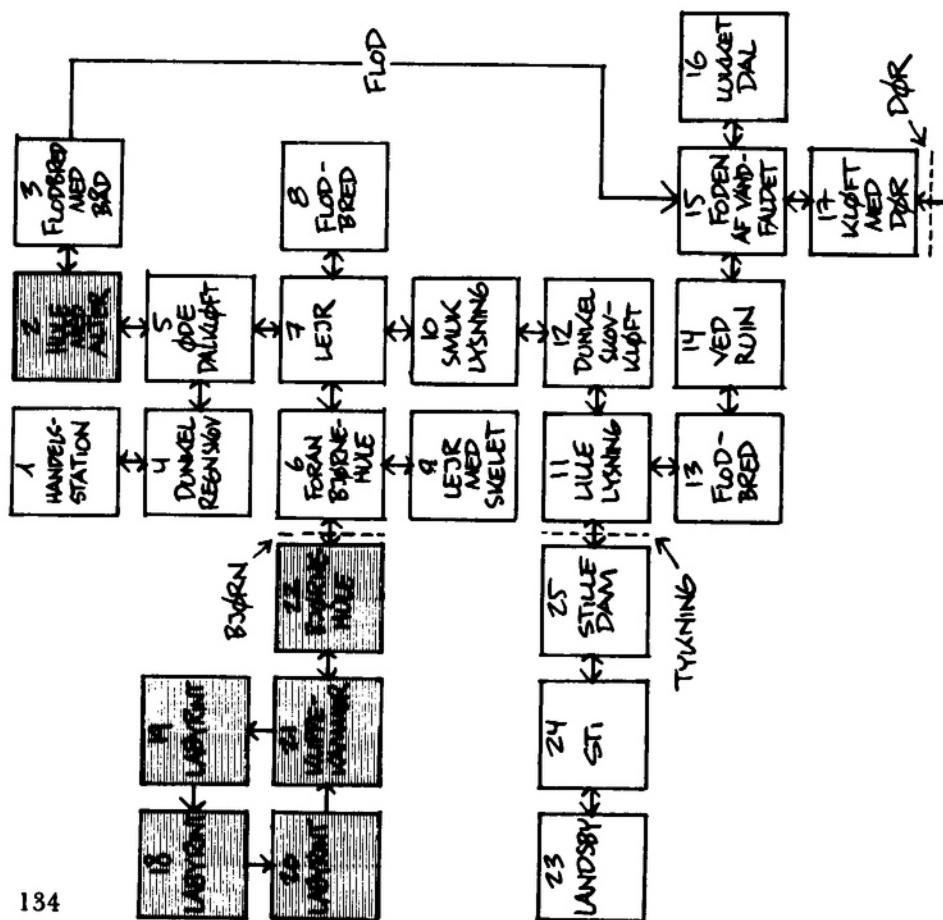
Med skatten i den ene hånd, lommelygten i den anden og nøglen i lommen begiver man sig tilbage mod udgangen, for

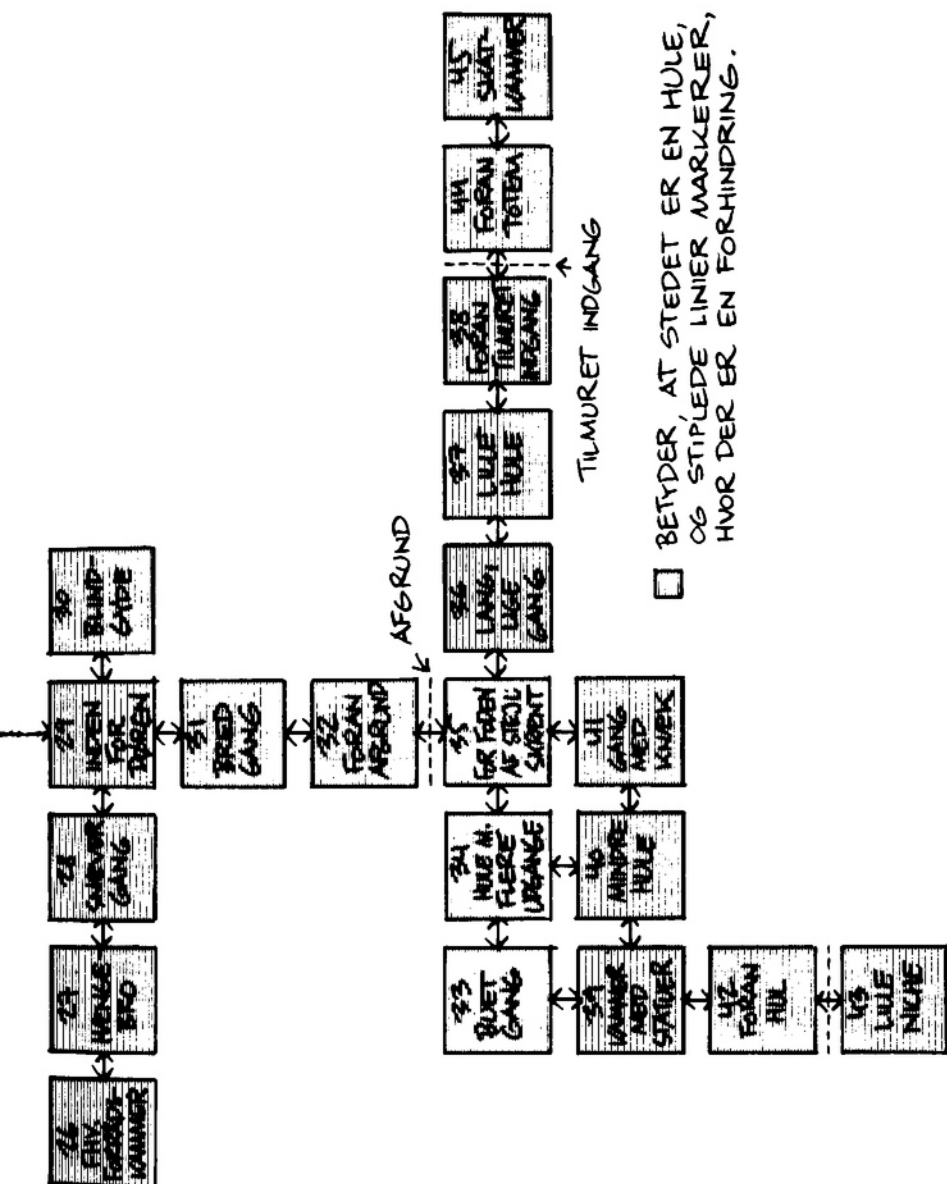


man har i begyndelsen af spillet fået at vide, at spillet først er forbi, når man er nået tilbage til sin lejr med skatten. Inden man når ud af hulen, er der en temmelig stor chance for, at tilfældighedstalsgeneratoren lader en drilagtig abe hugge skatten fra en

og anbringe den et andet (tilfældigt valgt) sted i hulen. Så bliver man nødt til at lede hulen igennem igen, indtil man finder skatten endnu en gang – og der er ingen garanti for, at aben ikke dukker op indtil flere gange igen – og imens tikker uret.

Hvis man overhovedet når ud af hulen igen, er der kun én hindring tilbage, før man når lejren – og den træder kun i kraft, hvis man undervejs har lagt riflen eller ammunitionen fra sig. Så sker der det, at man gentagne gange bliver angrebet af en blodtørstig panter og hele tiden bliver nødt til at flygte. Man kan faktisk slet ikke komme tilbage til lejren uden at skyde panteren. Men *kommer* man tilbage til lejren, er spillet forbi – og vel at





mærke vundet. Man kan begynde på et nyt, og det bliver garanteret ikke *helt* magen til det første!

På disse to sider ser du et kort over Amazonlandskabet, og herefter kommer en LISTning af spillet med en efterfølgende forklaring:

```

5 REM PA SKATTEJAGT I AMAZONLANDET *****
9 REM INDLEDNING *****
10 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****
30 PRINT "*****"
40 PRINT "    PA SKATTEJAGT I AMAZONLANDET"
50 PRINT:PRINT "eventyrspil for opdagelsesrejsende"
60 PRINT "*****"
70 FOR I=1 TO 10:PRINT:NEXT I
80 PRINT "HVAD HEDDER DU";
90 INPUT NA$
100 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****
105 PRINT "OKAY, ";NA$;"...":PRINT
110 PRINT "DU SKAL BRUGE BESTEMTE"
120 PRINT "KOMMANDOER FOR AT SPILLE"
130 PRINT "DETTE SPIL. NAR SPILLET ER"
140 PRINT "GAET I GANG, KAN DU HVER GANG"
150 PRINT "DU FAR SPØRGSMÅLET 'HVAD VIL"
160 PRINT "DU GØRE?' SKRIVE 'HJÆLP',"
170 PRINT "HVIS DU SKULLE VÆRE I TVIVL"
180 PRINT "OM NOGET."
190 PRINT:PRINT "SPILLET GAR UD PÅ AT FINDE"
200 PRINT "SKATTEN OG BRINGE DEN"
210 PRINT "TILBAGE INDEN FOR"
220 PRINT "DEN ANGIVNE TIDSFRIST,"
250 T=(VAL(RIGHT$(TIME$,2))+1)*
    (VAL(MID$(TIME$,4,2)))*(VAL(LEFT$(DATE$,2)));
    REM COMMODORE-BRUGER, STRYG DENNE LINIE *****
260 RANDOMIZE T:REM COMMODORE-BRUGER, STRYG DENNE
    LINIE *****
270 GOSUB 6590:REM NED TIL MELLEMRUMSTANGENTROUTINE
    *****
280 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****

```

```

285 PRINT "DU HAR I DIN AFDØDE ONKELS"
290 PRINT "GEMMER FUNDET ET BREV FRA"
300 PRINT "EN AF HANS VENNER, DER VAR"
310 PRINT "OPDAGELSEFREJSENDE. I BREVET"
320 PRINT "STAR, AT DER FINDES EN SKAT,"
330 PRINT "EN KOSTELIG SMARAGDKRONE,"
340 PRINT "I ET GADEFULDT TEMPEL MIDT"
350 PRINT "UDE I AMAZONJUNGLER. DER ER"
360 PRINT "EN LÆNGDE- OG BREDDEANGIVELSE"
370 PRINT "I BREVET, OG DU BEGYNDER"
380 PRINT "IVRIGT AT PAKKE DINE SAGER"
390 PRINT "FOR AT DRAGE PÅ SKATTEJAGT."
400 GOSUB 6590
410 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****
415 PRINT "I FLYET LÆSER DU TIL DIN"
420 PRINT "BEKYMRING I EN AVIS, AT"
430 PRINT "DET OMRÅDE, HVORI SKATTEN"
440 PRINT "BEFINDER SIG, VIL BLIVE"
450 PRINT "OVERSVØMMET OM 10 DAGE, NÅR"
460 PRINT "EN NYOPFØRT DÆMNING BLIVER"
470 PRINT "ÅBNET. DA FLYET LANDER, LEJER"
480 PRINT "DU EN HELIKOPTER, OG PILOTEN"
490 PRINT "SÆTTER DIG AF MIDT UDE I"
500 PRINT "AMAZONLANDET OG LOVER DIG"
510 PRINT "AT KOMME TILBAGE OG HENTE"
520 PRINT "DIG DEN 10. JULI KL. 12."
530 PRINT "HAN FORTÆLLER DIG OGSÅ, AT"
540 PRINT "DER I NÆRHEDEN ER EN"
550 PRINT "HANDELSSTATION, HVOR DU KAN"
560 PRINT "KØBE, HVAD DU HAR BRUG FOR."
570 PRINT "SÅ FLYVER HAN BORT."
580 GOSUB 6590:REM NED TIL MELLEMRUMSTANGENTRUTINE
    *****

```

```

590 PRINT:PRINT "OKAY. GOD FØRNUJELSE.":PRINT "VENT
    VENLIGST ET ØJEBLIK."
600 PRINT:PRINT "DET UNDERSTREGES, AT SPILLET"
610 PRINT "IKKE ER FÆRDIGT, FØR SKATTEN"
620 PRINT "ER BRAGT TILBAGE."
639 REM DATA *****
640 LO=7:MI=720:DA=1:TS=1:LE=1:SP=35
650 AU=15:AN=37:AL=45:GL=19
660 S$=CHR$(13)
670 C$="DU KAN IKKE BÆRE DEN ALENE."
680 D$="OM IGEN, "+NA$+"!"
690 E$="DET HAR DU ALLEREDE GJORT."
700 F$="DU RAMMER VED SIDEN AF."
710 G$="DET FAR DU IKKE NOGET UD AF."
720 H$="DU HAR IKKE NOGET AT ABNE MED."
730 J$="MAN KAN IKKE"+S$+"BRUGE NOGET, MAN IKKE
    HAR."
740 K$="DET BLIVER DU IKKE KLOGERE AF."
750 L$="DU HAR IKKE NOGE"
760 M$="DET KAN DU IKKE."
770 N$="DINE BÆRERE"
780 O$="ET GODT RAD:BEGYND FORFRA"
790 P$="HVILKEN NØGLE?" + S$ + "HVAD ER DEN LAVET AF?"
800 R$="DU BÆRER FØLGENDE:"
810 T$="HAR DU FLERE GENIALE FORSLAG?"
820 X$="STAR"
830 W$="ER"+S$
840 U$=X$+S$
850 DIM G(AN),G$(AN),U$(AU),N$(AN)
860 DIM V$(4),L$(AL),L(AL,8)
870 FOR I=1 TO GL:READ G(I):NEXT I
880 FOR I=1 TO GL:READ G$(I):NEXT I
890 FOR I=1 TO AU:READ U$(I):NEXT I
900 FOR I=1 TO AN:READ N$(I):NEXT I
910 FOR I=1 TO 4:READ V$(I):NEXT I

```

```

920 FOR I=1 TO AL:FOR J=1 TO 8:READ L(I,J):NEXT J:
    NEXT I
930 DATA 0,0,0,0,0,20,0,9,0,0,0,0,43,45,45,0,0,0,0
940 DATA EN LOMMELYGTE,NOGET PROVIANT,NOGET
    AMMUNITION
950 DATA EN SØLVNØGLE,EN KNIV,EN MACHETE,EN
    DASEABNER
960 DATA EN DAGBOG,EN GULDNØGLE,EN RIFFEL,NOGET
    DYNAMIT,ET TOV,EN JERNNØGLE
970 DATA EN FLASKE,EN SMARAGDKRONE,ET SKRIN,NOGLE
    TÆNDSTIKKER
980 DATA EN DØD LEOPARD,EN DØD BJØRN
990 DATA GA,SKY,ABN,TAG,LÆG,BRU,KØB,DRI
1000 DATA UND,SPI,LÆS,GEM,FOR,ANT,TØM:REM HVIS Æ=AE
    SA LÆS=STU *****
1010 DATA LOM,PRO,AMM,SØL,KNI,MAC,DAS,DAG,GUL,RIF,
    DYN
1020 DATA TOV,JER,FLA,SMA,SKR,TÆN,LEO,BJØ
1030 DATA IND,PAN,TRU,DØR,BAD,NØG,VAN,WHI,TYK,RUI,
    BRO,MAD,ETI,SPI
1040 DATA NOR,SYD,ØST,VES
1050 DATA NORDPA,SYDPA,ØSTPA,VESTPA
1060 LA$="FORSKELLIGE TING:"+S$+"AMMUNITION,
    PROVIANT,"+S$
1070 LB$="LOMMELYGTER, TÆNDSTIKKER"+S$+"OG DEN
    SLAGS."
1080 L$(1)=W$+"VED EN HANDELSSTATION,"+S$+"HVOR DU
    KAN KØBE"+S$+LA$+LB$
1090 DATA 0,4,0,0,0,30,0,0
1100 L$(2)=X$+" I EN HULE"+S$+"MED ET GAMMELT
    STENALTER."
1110 DATA 0,5,3,0,0,45,66,0
1120 L$(3)=U$+"VED EN FLODBRED,"+S$+"HVOR DER LIGGER
    EN BAD."
1130 DATA 0,0,0,2,0,0,0,66

```

1140 L\$(4)=W\$+"I EN DUNKEL REGNSKOV."
 1150 DATA 1,0,5,0,30,0,166,0
 1160 L\$(5)=U\$+"I EN ØDE DALKLØFT."
 1170 DATA 2,7,0,4,45,30,0,166
 1180 L\$(6)=U\$+"FORAN INDGANGEN TIL EN HULE."
 1190 DATA 0,9,7,0,0,30,52,2
 1200 L\$(7)=W\$+"I DIN LEJR MIDT UDE"+S\$+"I
 AMAZONJUNGLLEN."
 1210 DATA 5,10,8,6,30,24,120,52
 1220 L\$(8)=U\$+"VED BREDDEN AF EN FLOD."+S\$+"DU KAN
 IKKE KOMME VIDERE."
 1230 DATA 0,0,0,7,0,0,0,120
 1240 LC\$=S\$+"AF EN OPDAGELSESEKSESENDE."
 1250 L\$(9)=U\$+"I EN GAMMEL LEJR,"+S\$+"HVOR DU SER
 SKELETTET"+LC\$
 1260 DATA 6,0,0,0,30,0,0,0
 1270 L\$(10)=U\$+"I EN SMUK LYSNING."
 1280 DATA 7,12,0,0,24,58,0,0
 1290 LD\$=S\$+"AF EN TÆT TYKNING."
 1300 L\$(11)=W\$+"MIDT I EN LILLE LYSNING."+S\$+"MOD
 VEST SPÆRRES VEJEN"+LD\$
 1310 DATA 0,13,12,0,0,16,61,15
 1320 L\$(12)=U\$+"I EN DUNKEL SKOVKLØFT."
 1330 DATA 10,0,0,11,58,0,0,61
 1340 L\$(13)=W\$+"VED BREDDEN AF EN STOR FLOD."
 1350 DATA 11,0,14,0,16,0,125,0
 1360 L\$(14)=U\$+"VED EN GAMMEL RUIN NÆR FLODEN."
 1370 DATA 0,0,15,13,0,0,250,125
 1380 L\$(15)=U\$+"VED FODEN AF ET VANDFALD."
 1390 DATA 0,17,16,14,0,2,160,250
 1400 L\$(16)=U\$+"I EN ØDE OG LUKKET DAL."
 1410 DATA 0,0,0,15,0,0,0,160
 1420 L\$(17)="ER I EN KLØFT,"+S\$+"HVOR DER FOR
 ENDEN"+S\$+"ER EN AFLAST DØR."
 1430 DATA 15,0,0,0,2,2,0,0

1440 L\$(18)=W\$+"I EN FORVIRRENDE LABYRINT"+S\$+"AF
 SNÆVRE KLIPPEGANGE."
 1450 DATA 19,19,20,19,10,10,10,10
 1460 L\$(19)=L\$(18)
 1470 DATA 19,18,19,19,10,10,10,10
 1480 L\$(20)=L\$(18)
 1490 DATA 19,21,19,19,10,10,10,10
 1500 L\$(21)=U\$+"I ET LILLE KLIPPEKAMMER"+S\$+"BAG
 BJØRNENS HULE."
 1510 DATA 19,0,22,0,10,0,5,10
 1520 L\$(22)=U\$+"I BJØRNENS HULE."
 1530 DATA 0,0,6,21,0,0,2,5
 1540 LE\$="DER ER MEGET IVRIGE"+S\$+"EFTER AT TJENE
 PENGE,"+S\$
 1550 LF\$="SELV OM DE ER I FÆRD MED"+S\$+"AT FLYTTE
 FRA OMRÅDET."
 1560 L\$(23)=W\$+"I EN LANDSBY"+S\$+"MED VENLIGTSINDEDE
 INDFØDTE,"+S\$+LE\$+LF\$
 1570 DATA 0,0,24,0,0,0,306,0
 1580 L\$(24)=U\$+"PA EN STI MIDT I URSKOVEN."
 1590 DATA 0,0,25,23,0,0,230,306
 1600 L\$(25)=U\$+"VED EN STILLE DAM"+S\$+"MIDT UDE I
 SKOVEN."
 1610 DATA 0,0,11,24,0,0,15,230
 1620 L\$(26)=U\$+"I EN LILLE HULE, DER ENGANG"+S\$+"HAR
 TJENT SOM FORRADSKAMMER."
 1630 DATA 0,0,27,0,0,0,12,0
 1640 L\$(27)=U\$+"PA EN SVAJENDE HÆGEBRO,"+S\$+"DER ER
 LAVET AF TOVVERK."
 1650 DATA 0,0,28,26,0,0,10,12
 1660 L\$(28)=W\$+"I EN SNÆVER GANG."
 1670 DATA 0,0,29,27,0,0,10,10
 1680 L\$(29)=U\$+"VED DØREN UD TIL JUNGLEN,"+S\$+"DØREN
 ER LAST."
 1690 DATA 17,31,30,28,2,27,25,10

1700 L\$(30)=W\$+"ENDT I EN BLINDGYDE."
 1710 DATA 0,0,0,29,0,0,0,25
 1720 L\$(31)=W\$+"I EN BRED GANG."
 1730 DATA 29,32,0,0,27,15,0,0
 1740 L\$(32)=U\$+"PA KANTEN AF EN AFGRUND."
 1750 DATA 31,0,0,0,15,2,0,0
 1760 L\$(33)=W\$+"I EN BUET GANG."
 1770 DATA 0,39,34,0,0,16,8,0
 1780 L\$(34)="ER MIDT I"+S\$+"EN HULE MED FLERE
 UDGANGE."
 1790 DATA 0,40,35,33,0,12,5,8
 1800 L\$(35)=U\$+"FOR FODEN AF EN STEJL SKRÆNT."
 1810 DATA 0,41,36,34,2,8,14,5
 1820 L\$(36)=W\$+"I EN LANG, LIGE GANG."
 1830 DATA 0,0,37,35,0,0,20,14
 1840 L\$(37)=U\$+"I EN LILLE HULE."
 1850 DATA 0,0,38,36,0,0,120,20
 1860 L\$(38)=U\$+"FORAN EN TILMURET INDGANG"+S\$+"TIL
 GUDERNE MA VIDE HVAD."
 1870 DATA 0,0,0,37,0,0,1,120
 1880 L\$(39)=U\$+"I ET LILLE KAMMER"+S\$+"MED NOGLE
 GADEFULDE STATUER."
 1890 DATA 33,42,40,0,16,20,8,2
 1900 L\$(40)=W\$+"I EN MINDRE HULE"+S\$+"MED TRE
 FORSKELLIGE UDGANGE."
 1910 DATA 34,0,41,39,12,0,3,8
 1920 L\$(41)=U\$+"MIDT I EN GANG,"+S\$+"DER SLAR ET
 VINKELRET KNÆK."
 1930 DATA 35,0,0,40,8,0,0,3
 1940 L\$(42)=U\$+"FOR ENDEN AF EN GANG,"+S\$+"DU SER ET
 HUL I VÆGGEN."
 1950 DATA 39,0,0,0,20,1,0,0
 1960 L\$(43)=U\$+"I EN LILLE NICHE."
 1970 DATA 42,0,0,0,1,0,0,0
 1980 L\$(44)=U\$+"FORAN EN SKRÆKINDJAGENDE TOTEM."

```

1990 DATA 0,0,45,38,0,3,2,1
2000 L$(45)=U$+"I SKATKAMMERET."
2010 DATA 0,0,0,44,0,0,0,1
2020 G(7)=INT(RND(1)*41)+2:REM DASEABNER PLACERES
      *****
2030 G(16)=INT(RND(1)*24)+2:REM SKRIN PLACERES *****
2040 G(10)=INT(RND(1)*16)+2:IF G(10)=9 THEN 2040:REM
      RIFFEL PLACERES *****
2050 G(11)=INT(RND(1)*41)+2:REM DYNAMIT PLACERES
      *****
2060 G(5)=INT(RND(1)*41)+2:REM KNIV PLACERES *****
2079 REM RENDRINGER *****
2080 AP$=S$
2090 IF LE=LO THEN 2110: REM SA MAN IKKE KAN FJERNE
      LEOPARDEN VED AT RYDDE SKÆRMEN
2100 IF LE<>0 THEN LE=INT(RND(1)*22)+1:IF LE=7 OR
      LE=6 OR LE=11 THEN 2100
2110 IF LO>25 AND G(15)=-1 THEN TS=1
2120 AA$=S$+"MEN STEDET ER VED AT"+S$+"BLIVE
      EVAKUERET OG"+S$
2130 AB$="UDBUDDET ER SPARSOMT."+S$
2140 IF G(1)=0 AND G(2)=0 AND G(3)=0 AND G(17)=0 AND
      LO=1 THEN AP$=AA$+AB$
2150 IF G(15)=-1 AND TS=1 THEN TR=INT(RND(1)*17)+26
2160 IF TS=3 THEN 2180
2170 IF LO<26 AND G(15)=-1 THEN TS=2
2180 IF (LO=2 OR (LO>17 AND LO<23) OR LO>25) AND
      G(1)<>-1 THEN LM=1:GOTO 2200
2190 LM=0
2200 IF VF=1 THEN G$(14)="EN TOM FLASKE"
2210 IF BD=1 THEN L$(3)=U$+"VED DEN FLODBRED"+S$+
      "HVOR DU TOG BADEN."
2220 IF BJ=1 THEN L$(6)=X$+"FORAN"+S$+"INDGANGEN
      TIL BJØRNEHULEN."

```

```

2230 IF TY=1 THEN L$(11)=U$+"MIDT I EN LILLE
      LYSNING."
2240 IF RE=1 THEN L(32,2)=35 AND L(35,1)=32
2250 IF HU=1 THEN L(42,2)=43:L$(42)=U$+"VED DET ABNE
      STENPANEL."
2260 IF IN<>1 THEN 2300
2270 L(17,2)=29:L(29,1)=17
2280 L$(17)=X$+" UDEN FOR"+S$+"INDGANGEN TIL
      SKATTEHULEN."
2290 L$(29)=X$+" INDEN FOR"+S$+"DØREN TIL
      SKATTEHULEN."
2300 AC$=S$+"DINE FORSKRÆMTE BÆRERE"+S$+"HAR LAST
      DØREN UDEFRA"+S$+AD$
2310 AD$="MED SØLVNØGLEN, SOM DE"+S$+"(BEKLAGELIGT
      NOK) TOG MED."+S$
2320 IF BA=2 AND LO=29 THEN IN=0:L(29,1)=0:AP$=AC$
2330 IF TV<>1 THEN 2350
2340 L(38,3)=44:L$(38)=U$+"VED DET STED HVOR
      DU"+S$+"BRUGTE DYNAMITTEN."
2350 IF BA<>1 OR LO<>44 THEN 2430
2360 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
2365 GOSUB 8450:REM UR-RUTINE *****
2370 PRINT:PRINT "DU "+X$+" FORAN":PRINT "EN
      SKRÆKINDJAGENDE TOTEM."
2380 PRINT "DA ";N$;" SER TOTEMEN,":PRINT "SMIDER DE
      (NÆSTEN) ALT,"
2390 PRINT "HVAD DE BÆRER, OG FLYGTER.":N=6:GOSUB
      9240
2400 BA=2:BE=0:G(4)=-2:FOR I=1 TO GL:IF G(I)=-1 THEN
      G(I)=LO
2410 NEXT I
2420 LM=1
2430 GOSUB 7880:GOTO 2450:REM SKÆRMTEKSTRUTINE *****
2439 REM STRENGDELING *****

```

```

2450 PRINT:PRINT "HVAD VIL DU GØRE";
2460 INPUT B$
2470 PRINT
2480 IF B$="R" THEN MI=MI+1:GOTO 2080
2490 IF B$<>"SOVE" THEN 2510
2500 PRINT "DU PRØVER,":PRINT "MEN DU KAN IKKE FALDE
      I SØVN.":GOTO 2450
2510 IF B$="N" THEN B$="GA NORDPA"
2520 IF B$="S" THEN B$="GA SYDPA"
2530 IF B$="Ø" THEN B$="GA ØSTPA"
2540 IF B$="V" THEN B$="GA VESTPA"
2550 IF B$="HJÆLP" THEN 6080
2560 IF B$="SEJLE" THEN 5950
2570 IF B$="STATUS" THEN 6650
2580 IF B$="FLYGTE" THEN 3070
2590 IF B$="STOP" OR B$="STOPPE" THEN 7750
2600 FOR I=1 TO LEN(B$)
2610 IF MID$(B$,I,1)=" " THEN 2640
2620 NEXT I
2630 GOTO 9300:REM OM IGEN HVIS KUN EET ORD *****
2640 U0$=LEFT$(B$,I-1)
2650 NO$=RIGHT$(B$, (LEN(B$)-I))
2660 IF U0$="GA" THEN 2720:REM IKKE HVIS A=AA *****
2670 IF LEN(U0$)<3 THEN 9300
2680 U0$=LEFT$(U0$,3):FOR I=1 TO AU:IF U0$=U$(I)
      THEN 2730
2690 NEXT I
2700 IF U0$="DRÆ" THEN PRINT "HVORDAN?":GOTO
      2450:REM HVIS Æ=AE SA DRA *****
2710 U0=0:GOTO 9300
2720 U0=1:GOTO 2740:REM IKKE HVIS A=AA *****
2730 U0=I
2740 IF LEN(NO$)<3 THEN 9300
2750 NO$=LEFT$(NO$,3):FOR I=1 TO AN:IF NO$=N$(I)
      THEN 2810

```

```

2760 NEXT I
2770 IF NO$="BER" THEN NO=20:GOTO 2830:REM # *****
2780 IF NO$="HEN" THEN NO=30:GOTO 2830:REM # *****
2790 IF NO$="SKA" OR NO$="KRO" THEN NO=15:GOTO 2830
2800 NO=0:GOTO 9300
2810 NO=I
2829 REM UDSAGNSORD *****
2830 IF UO<8 THEN 2850
2840 IF UO>7 THEN 2860
2850 ON UO GOTO 2880,3320,3630,3880,4130,4230,4640
2860 ON UO-7 GOTO 4950,5070,5520,5770,6720,7110,
      7440,7610
2879 REM GA *****
2880 IF LE<>LO AND TR<>LO AND LO<>6 THEN 2970
2890 IF BJ=1 AND LO=6 THEN 2970
2900 IF LO>25 THEN 2970
2910 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
2920 PRINT "DU KAN IKKE GA NOGEN STEDER.":PRINT "DER
      ER NOGET, DER TRUER DIG,"
2930 PRINT "OG DU ER NØDT TIL AT FLYGTE":PRINT
      "ELLER USKADELIGGØRE TRUSLEN."
2940 PRINT "(SELV OM DET IKKE ER SIKKERT,":PRINT "AT
      DU SLIPPER GODT FRA DET)"
2950 IF G(1)=LO AND (LM=1 OR LM=2) THEN PRINT "ET
      GODT TIP: TAG LOMMELYG TEN!"
2960 GOTO 2450
2970 IF NO<34 THEN 9300
2980 NO=NO-33
2990 IF LO<>32 OR NO<>2 THEN 3030
3000 PRINT "DU STYRTER I AFGRUNDEN":PRINT "OG SLAR
      DIG IHJEL."
3010 PRINT "JEG SAGDE JO, AT DU KUN":PRINT "KUNNE GA
      MOD NORD, DIT FÆ!"
3020 GOTO 9080

```

```

3030 IF L(LO,NO)=0 THEN PRINT "DU KAN IKKE GA DEN
      VEJ.":GOTO 2450
3040 MI=MI+L(LO,(NO+4))+BE+5
3050 LO=L(LO,NO):GOTO 2080
3069 REM FLYGTE *****
3070 IF (LE<>LO AND TR<>LO AND LO<>6) OR (LO=6 AND
      BJ=1) THEN 3090
3080 GOTO 3100
3090 PRINT "HVORFOR DOG DET?":GOTO 2450
3100 FL=INT(RND(1)*9)+1
3110 IF FL<4 THEN 3290
3120 PRINT "DU SMIDER ALT, HVAD DU HAR,"
3130 PRINT "OG FLYGTER I VILD PANIK":PRINT "UDEN AT
      TÆNKE OVER"
3140 PRINT "I HVILKEN RETNING DU LØBER."
3150 FOR I=1 TO GL:IF G(I)=-1 THEN G(I)=LO
3160 NEXT I
3170 BE=0:AY=LO
3180 AX=INT(RND(1)*4)+1
3190 MI=MI+L(AY,(AX+4))*5
3200 LO=L(LO,AX)
3210 IF LO=0 THEN LO=AY:GOTO 3180
3220 IF BA=1 THEN 3240
3230 GOTO 3280
3240 BA=2:BF=INT(RND(1)*25)+1
3250 IF BF=LO THEN 3240
3260 IF LO>25 THEN G(4)=-2
3270 PRINT N$;" ER PIST VÆK."
3280 N=5:GOSUB 9240:GOTO 2080
3290 PRINT "DU PRØVER AT FLYGTE, MEN"
3300 PRINT "BLIVER OVERMANDET OG DRÆBT.":GOTO 9080
3319 REM SKYDE *****
3320 IF G(10)=-1 THEN 3340
3330 PRINT L$;"T AT SKYDE MED.":GOTO 2450
3340 IF G(3)<>-1 THEN PRINT L$;"T AMMUNITION.":GOTO

```

```

2450 3350 IF NO<>19 THEN 3400
3360 IF LO<>6 THEN PRINT "DER ER INGEN BJØRN
      HER.":GOTO 2450
3370 IF BJ=1 THEN PRINT E$:GOTO 2450
3380 PRINT "BJØRNEN FALDER DØD OM,":PRINT "OG DU KAN
      GA IND I DENS HULE."
3390 BJ=1:G(19)=6:L(6,4)=22:N=2:GOSUB 9240:GOTO 2080
3400 IF NO<>18 THEN 3500
3410 IF LE=0 THEN PRINT E$:GOTO 2450
3420 IF LE<>LO THEN PRINT "DER ER INGEN LEOPARD
      HER.":GOTO 2450
3430 KU=INT(RND(1)*3)+1
3440 IF KU>1 THEN 3460
3450 PRINT "DU HAR SKUDT EN LEOPARD.":G(18)=LO:
      LE=0:GOTO 2450
3460 PRINT F$:PRINT "LYDEN AF SKUDET FAR":PRINT
      "LEOPARDEN TIL AT FLYGTE."
3470 LE=INT(RND(1)*25)+1
3480 IF LE=LO THEN 3470
3490 GOTO 2450
3500 IF NO=20 AND (LO=23 OR BA=1) THEN 3520
3510 GOTO 3550
3520 PRINT "DE TAGER DIG DET MEGET ILDE OP."
3530 PRINT "DA DU HAR SKUDT TRE AF DEM,":PRINT
      "OVERMANDER RESTEN DIG"
3540 PRINT "OG SLAR DIG IHJEL.":GOTO 9080
3550 IF NO<>21 THEN 3600
3560 PRINT "DU RAMMER VED SIDEN AF.":PRINT "LYDEN AF
      SKUDET"
3570 PRINT "FAR PANTEREN TIL AT FLYGTE."
3580 TR=INT(RND(1)*5)+11:IF TR=LO THEN 3580
3590 GOTO 2450
3600 IF NO=22 THEN PRINT "DU MA PRÆCISERE TRUSLEN.":
      GOTO 2450

```



```

3610 PRINT G#:GOTO 2450
3629 REM ABNE *****
3630 IF NO<>16 THEN 3680
3640 IF G(9)<>-1 THEN 3670
3650 PRINT "DU ABNER SKRINET":PRINT "MED GULDNØGLEN
      OG SER,"
3660 PRINT "AT DET INDEHOLDER":PRINT "EN STOR
      SØLVNØGLE.":G(4)=LO:GOTO 2450
3670 PRINT H#:GOTO 2450
3680 IF NO<>18 THEN 3730
3690 IF G(NO)=LO OR G(NO)=-1 THEN 3710
3700 GOTO 3720
3710 PRINT "SLET IKKE NOGEN DUM IDE.":PRINT "MEN
      HVORDAN?":GOTO 2450
3720 PRINT "HA! HA! HA!":GOTO 2450
3730 IF NO<>23 THEN 3860
3740 IF LO<>17 THEN 3810
3750 IF IN=1 THEN PRINT E#:GOTO 2450
3760 IF G(4)=-1 THEN 3780
3770 GOTO 3800
3780 PRINT "DØREN GLIDER KNIRKENDE OP":PRINT "PA
      SINE RUSTNE HÆNGSLER."
3790 IN=1:N=2:GOSUB 9240:GOTO 2080
3800 PRINT H#:GOTO 2450
3810 IF LO<>29 THEN PRINT "DER ER IKKE NOGEN DØR
      HER.":GOTO 2450
3820 IF IN=1 THEN PRINT E#:GOTO 2450
3830 IF G(13)<>-1 THEN PRINT H#:GOTO 2450
3840 BA=3:IN=1:PRINT "DU ABNER DØREN MED JERNNØGLEN"
3850 PRINT "OG HAR ATTER ADGANG TIL JUNGLEN.":N=3:
      GOSUB 9240:GOTO 2080
3860 GOTO 9300
3879 REM TAGE *****
3880 IF LO=3 AND NO=24 THEN 3900

```

```

3890 GOTO 3920
3900 PRINT "HVORFOR SKULLE DU DOG":PRINT "SLÆBE
      RUNDT PA DEN?"
3910 GOTO 9300
3920 IF NO=25 THEN 4150
3930 IF NO=12 AND LO=27 AND G(12)=0 THEN 3950
3940 GOTO 3970
3950 PRINT "JA...MEN DU MA HELLERE UNDERSØGE,":PRINT
      "OM DER ER NOGET."
3960 GOTO 2450
3970 IF G(NO)=-1 THEN PRINT E$:GOTO 2450
3980 IF LO=1 AND G(NO)=0 THEN 4000
3990 GOTO 4010
4000 PRINT M$:PRINT "DU MA PÆNT BETALE.":GOTO 2450
4010 IF G(NO)<>LO THEN PRINT "MEGET MORSOMT.":GOTO
      2450
4020 IF BA=1 THEN 4100
4030 IF (NO=3 AND BE>3) OR BE>7 THEN PRINT "DU BÆRER
      PA FOR MEGET.":GOTO 2450
4040 IF NO=3 THEN BE=BE+5:GOTO 4090
4050 IF NO=2 OR NO=11 OR NO=18 OR NO=19 THEN 4070
4060 GOTO 4080
4070 PRINT C$:GOTO 2450
4080 BE=BE+1
4090 IF LO=35 AND NO=12 THEN RE=2
4100 G(NO)=-1:PRINT "UDMÆRKET.":N=.5:GOSUB 9240
4110 GOTO 2080
4129 REM LÆGGE *****
4130 IF NO=25 THEN 4150
4140 GOTO 4160
4150 PRINT P$:GOTO 2450
4160 IF G(NO)=-1 THEN 4180
4170 PRINT "FOR AT KUNNE LÆGGE NOGET":PRINT "MA MAN
      HAVE DET.":GOTO 2450
4180 IF NO=3 THEN BE=BE-5:GOTO 4200

```

```

4190 BE=BE-1
4200 PRINT "SOM DU VIL.":N=.5:GOSUB 9240:G(N0)=LO
4210 GOTO 2080
4229 REM BRUGE *****
4230 IF NO<>6 THEN 4310
4240 IF G(6)<>-1 THEN PRINT J$:GOTO 2450
4250 IF LO=27 THEN 4330
4260 IF LO<>11 THEN 4290
4270 PRINT "DU HUGGER DIG":PRINT "IGENNEM
      TYKNINGEN."
4280 MI=MI+120:LO=25:L(11,4)=25:TY=1:N=2:GOSUB
      9240:GOTO 2080
4290 IF G(18)=-1 OR G(18)=LO THEN 4360
4300 GOTO 9300
4310 IF NO<>5 THEN 4480
4320 IF G(5)<>-1 THEN PRINT J$:GOTO 2450
4330 IF LO<>27 THEN 4360
4340 PRINT "DU SKÆRER TOVVÆRKET OVER":PRINT "OG
      HÆNGBROEN STYRTER"
4350 PRINT "I DYBET. DU BLIVER DRÆBT.":GOTO 9080
4360 IF G(9)<>0 THEN 4420
4370 IF G(18)<>-1 THEN 4390
4380 PRINT "LÆG VENLIGST LEOPARDEN":PRINT "PÆNT NED
      PÅ JORDEN.":GOTO 2450
4390 IF G(18)<>LO THEN 4420
4400 PRINT "DU SKÆRER LEOPARDEN OP":PRINT "OG FINDER
      EN LILLE GULDNØGLE"
4410 PRINT "I DENS MAVE.":G(9)=LO:GOTO 2450
4420 IF NO<>5 THEN 4460
4430 PRINT "DU GAR LIDT FOR RASK":PRINT "TIL VÆRKS,
      OG KNIVEN KNÆKKER."
4440 PRINT "DET VAR BARE ÆRGERLIGT.":PRINT "NU HAR
      DU IKKE DEN LÆNGERE."
4450 G(5)=0:BE=BE-1:GOTO 2450

```

```

4460 IF NO=6 THEN 4300
4470 GOTO 2450
4480 IF NO<>12 THEN 4620
4490 IF G(12)<>-1 AND LO<>32 AND LO<>35 THEN PRINT
      J$:GOTO 2450
4500 IF LO<>32 AND LO<>35 THEN PRINT G$:GOTO 2450
4510 IF G(12)<>LO AND G(12)<>-1 THEN PRINT "HVAD FOR
      ET TOV?":GOTO 2450
4520 IF LO=35 THEN 4550
4530 PRINT "DU FIRER DIG NED I AFGRUNDEN":PRINT "VED
      HJÆLP AF TOVET."
4540 G(12)=35:RE=1:LO=35:MI=MI+15:BE=BE-1:N=2:GOSUB
      9240:GOTO 2080
4550 IF RE<>2 THEN 4590
4560 PRINT M$:PRINT "DER ER ALT FOR HØJT OP."
4570 PRINT "DET VAR DUMT AF DIG AT TAGE DET."
4580 PRINT "NU KOMMER DU ALDRIG UD AF HULEN.":GOTO
      2450
4590 IF RE<>1 THEN 4620
4600 G(12)=32:LO=32:BE=BE+1:MI=MI+15:PRINT "DU MÅ
      LÆGGE KRÆFTER I"
4610 PRINT "FOR AT KOMME OP AD TOVET.":N=2:GOSUB
      9240:GOTO 2080
4620 GOTO 9300
4639 REM KØBE *****
4640 IF LO=23 THEN 4670
4650 IF LO<>1 THEN PRINT "DU KAN IKKE KØBE NOGET
      HER.":GOTO 2450
4660 GOTO 4750
4670 IF NO=20 AND BA<>0 THEN 4690
4680 GOTO 4710
4690 PRINT "DEM, DU LEJEDE, VAR DE SIDSTE"
4700 PRINT "BÆRERE, DER VAR TIL RADIGHED.":GOTO 2450
4710 IF NO=20 AND BA=0 THEN 4730
4720 GOTO 4740

```

```

4730 PRINT "DU HAR NU EN FLOK":PRINT "INDFØDTE
      BÆRERE.":BA=1:GOTO 2450
4740 PRINT "PRØV AT KØBE NOGET ANDET.":GOTO 2450
4750 IF (NO>3 AND NO<>17) OR G(NO)<>0 THEN 4770
4760 GOTO 4780
4770 PRINT M$:PRINT "DER ER UDSOLGT.":GOTO 2450
4780 IF BA=0 AND NO=2 THEN 4800
4790 GOTO 4820
4800 PRINT "PROVANTEN ER MEGET TUNG.":PRINT C$
4810 BE=BE-1:G(2)=1:N=2:GOSUB 9240:GOTO 2080
4820 G(NO)=-1
4830 IF NO=3 THEN BE=BE+5:GOTO 4850
4840 BE=BE+1
4850 IF BA=0 AND BE>8 THEN 4870
4860 GOTO 4890
4870 PRINT "DU BÆRER PA FOR MEGET, SA DER"
4880 PRINT "ER INGEN IDE I AT KØBE DET.":GOTO 4900
4890 PRINT "UDMÆRKET.":N=1:GOSUB 9240:GOTO 2080
4900 G(NO)=0
4910 IF NO=3 THEN BE=BE-5:GOTO 4930
4920 BE=BE-1
4930 GOTO 2450
4949 REM DRIKKE *****
4950 IF NO=26 THEN PRINT "DU ER IKKE TØRSTIG.":GOTO
      2450
4960 IF NO<>27 THEN 5050
4970 IF G(14)<>-1 AND G(14)<>LO THEN PRINT "HVOR ER
      DEN?":GOTO 2450
4980 IF VF=1 THEN PRINT "FLASKEN ER TOM.":GOTO 2450
4990 PRINT "DU DRIKKER":PRINT "EN HEL FLASKE WHISKY"
5000 PRINT "OG FALDER BEVIDSTLØS OM."
5010 PRINT "DU SOVER I ATTEN TIMER":PRINT "(DET
      KALDER JEG TIDSSPILDE)"
5020 PRINT "OG DA DU VAGNER, ER DU MERE TRÆT":PRINT
      "END FØR DU DEJSEDE OM."

```

```

5030 PRINT Q$;"!"
5040 MI=MI+1080:SO=0:VF=1:N=7:GOSUB 9240:GOTO 2080
5050 PRINT "DET MENER DU DA":PRINT "VIST IKKE
      ALVORLIGT.":GOTO 2450
5069 REM UNDERSØGE *****
5070 IF NO>30 THEN PRINT K$:GOTO 2450
5080 IF NO=18 THEN 5160
5090 IF NO<20 AND (G(NO)<>-1 AND G(NO)<>LO) THEN
      9300
5100 IF NO<9 THEN 5140
5110 IF NO>8 AND NO<17 THEN 5150
5120 IF NO>16 AND NO<25 THEN 5160
5130 IF NO>24 THEN 5170
5140 ON NO GOSUB 5500,5180,5500,5190,5500,5500,5500,
      5200:GOTO 2450
5150 ON NO-8 GOSUB 5500,5210,5500,5230,5190,5250,
      5260,5270:GOTO 2450
5160 ON NO-16 GOSUB 5500,5290,5500,5500,5500,5500,
      5340,5500:GOTO 2450
5170 ON NO-24 GOSUB 5370,5500,5500,5380,5400,5430:
      GOTO 2450
5180 PRINT "DEN BESTAR AF KONSERVESDASER.":RETURN
5190 PRINT "PA NØGLEN STAR:":PRINT "'DER FINDES EN
      KOPI AF MIG.'":RETURN
5200 PRINT "DER STAR NOGET I DEN.":RETURN
5210 IF G(3)<>-1 THEN PRINT "DEN ER IKKE LADT.":
      RETURN
5220 PRINT "DEN ER SKUDKLAR.":RETURN
5230 IF G(12)=0 AND LO=27 THEN PRINT "PRØV AT
      UNDERSØGE BROEN.":RETURN
5240 GOTO 5500
5250 PRINT "DER ER EN ETIKETTE PA FLASKEN.":RETURN
5260 PRINT "DEN ER UTROLIG KOSTBAR.":RETURN
5270 IF G(4)≠0 THEN PRINT "DET RASLER, NAR DU RYSTER
      DET.":RETURN

```

```

5280 PRINT "DET ER TOMT.":RETURN
5290 IF G(18)<>0 THEN 5310
5300 PRINT "DU BLIVER NØDT TIL":PRINT "AT SKYDE DEN
      FØRST.":RETURN
5310 IF G(9)<>0 THEN 5330
5320 PRINT "UDADTIL LIGNER DEN":PRINT "ENHVER ANDEN
      LEOPARD, MEN...":RETURN
5330 GOTO 5500
5340 IF (LO=17 OR LO=29) AND IN=0 THEN 5360
5350 GOTO 5500
5360 PRINT "DEN KAN KUN":PRINT "ABNES MED EN
      NØGLE.":RETURN
5370 PRINT P$:RETURN
5380 IF LO=11 AND L(11,4)=0 THEN PRINT "DU KAN HUGGE
      DIG IGENNEM DEN.":RETURN
5390 GOTO 5500
5400 IF LO<>14 THEN 5420
5410 PRINT "DER STAR 'KILROY WAS HERE'":PRINT "PA EN
      AF MURENE.":RETURN
5420 PRINT D$:RETURN
5430 IF LO=27 AND G(12)=0 THEN 5450
5440 GOTO 5460
5450 PRINT "DER HÆNGER ET LANGT":PRINT "LØST TOV NED
      FRA DEN.":G(12)=27:RETURN
5460 IF LO=27 AND G(12)<>0 THEN 5480
5470 GOTO 5490
5480 PRINT "DER ER IKKE LÆNGERE":PRINT "NOGET
      USÆDVANLIGT VED DEN.":RETURN
5490 PRINT D$:RETURN
5500 PRINT K$:RETURN
5519 REM SPISE *****
5520 IF NO=31 THEN PRINT "GOD IDE, HAR DU NOGET?":
      GOTO 2450
5530 IF G(NO)<>-2 THEN 5560

```



```
5540 PRINT "DU HAR ALLEREDE":PRINT "SPIST DEN, DIN  
      #DEDOLK!"  
5550 PRINT "DER ER IKKE MERE TILBAGE.":GOTO 2450  
5560 IF NO<>2 THEN 5620  
5570 IF G(NO)<>-1 AND G(NO)<>LO THEN 5590  
5580 GOTO 5600  
5590 PRINT L$;"T PROVIANT.":GOTO 2450  
5600 IF G(7)<>-1 THEN PRINT L$+"T":PRINT "AT ABNE  
      DEN MED.":GOTO 2450  
5610 GOTO 5730  
5620 IF NO<>18 AND NO<>19 THEN 5750  
5630 IF (G(NO)<>-1 AND G(NO)<>LO) OR G(NO)=0 THEN  
      5650
```



```

5640 GOTO 5660
5650 PRINT T$:GOTO 2450
5660 IF G(5)<>-1 AND G(6)<>-1 THEN 5680
5670 GOTO 5690
5680 PRINT L$+"T":PRINT "AT PARTERE DEN MED.":GOTO
      2450
5690 IF NO=18 AND G(9)=0 THEN 5710
5700 GOTO 5730
5710 PRINT "DA DU SKÆRER LEOPARDEN OP,":PRINT
      "FINDER DU EN LILLE GULDØGLE"
5720 PRINT "I MAVEN PÅ DEN.":PRINT:G(9)=LO
5730 G(NO)=-2:AZ=DA+9:SP=36:PRINT "DU STILLER DIN
      SULT."
5740 BE=BE-2:MI=MI+30:N=3:GOSUB 9240:GOTO 2080
5750 PRINT "DU ER ET VITTIKT HOVED, HVA'?:GOTO 2450
5769 REM LÆSE ***** HVIS AE SA STUDERE
5770 IF NO<>8 THEN 5890
5780 IF G(8)<>-1 THEN PRINT L$+"N DAGBOG.":GOTO 2450
5790 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
5795 PRINT "I DAGBOGEN STAR BL.A. : "
5800 PRINT:PRINT "'26. AUGUST":PRINT
5810 PRINT "I DAG FANDT JEG EN LILLE":PRINT
      "GULDØGLE, SOM JEG NU"
5820 PRINT "GAR MED I EN KÆDE OM HALSEN.":PRINT "JEG
      VAR VED AT FARE"
5830 PRINT "VILD I HULEN, MEN SA KOM JEG":PRINT "TIL
      AT TÆNKE PÅ 'SØS'."
5840 PRINT "JEG HAR DESVÆRRE":PRINT "TABT MIN
      RIFFEL"
5850 PRINT "OG NU FRYGTER JEG":PRINT "FOR MIT LIV,
      FOR DER"
5860 PRINT "ER EN STOR LEOPARD,":PRINT "DER BLIVER
      VED MED AT KREDSE"
5870 PRINT "OM LEJREN. HVIS JEG IKK ...'"

```

```

5880 PRINT:PRINT "SA VIDT DEN":PRINT
      "OPDAGELSESEJSENDES DAGBOG.":GOTO 2450
5890 IF NO<>32 THEN 5920
5900 IF G(14)<>-1 AND G(14)<>LO THEN PRINT
      "HVA'BA?":GOTO 2450
5910 PRINT "PA ETIKETTEN STAR:":PRINT "SKOTSK
      WHISKY.":GOTO 2450
5920 PRINT "DER STAR 'MADE IN JAPAN'."
5930 PRINT T$:GOTO 2450
5949 REM SEJLE *****
5950 IF LO<>3 OR BD=1 THEN PRINT "I HVAD?":GOTO 2450
5960 SE=INT(RND(1)*10)+1
5970 DU$="DU "
5980 I$="DU "
5990 IF BA=1 THEN DU$="DU OG "+N$+" ":I$="I "
6000 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
6005 PRINT DU$;"SEJLER NED AD":PRINT "FLODEN OG
      BLIVER HURTIGT"
6010 PRINT "FANGET AF EN RIVENDE STRØM."
6020 IF SE>5 THEN 6050
6030 PRINT I$;"RYGER NED AD ET VANDFALD,":PRINT "OG
      BADEN KNUSES, MEN "
6040 PRINT I$;"OVERLEVER.":BD=1:MI=MI+30:LO=15:N=5:
      GOSUB 9240:GOTO 2080
6050 PRINT I$;"BLIVER KNUST MOD KLIPPERNE":PRINT
      "FOR FODEN AF ET VANDFALD."
6060 GOTO 9080
6079 REM HJÆLP *****
6080 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
6085 PRINT "DU KAN BRUGE"
6090 PRINT "FØLGENDE KOMMANDOER:"
6100 PRINT:PRINT "GA (NORDPA ETC) - MEN"
6110 PRINT "HER KAN DU OGSA NØJES MED"

```

```

6120 PRINT "AT SKRIVE 'N' FOR 'GA NORDPA',"
6130 PRINT "'S' FOR 'GA SYDPA' ETC."
6140 PRINT:PRINT "TAGE - LÆGGE - BRUGE - KØBE"
6150 PRINT "- SKYDE - ÅBNE - LÆSE -":REM HVIS ÅE SÅ
      STUDERE *****
6160 PRINT "UNDERSØGE - SPISE - DRIKKE"
6170 PRINT "- TØMME - ANTÆNDE"
6180 PRINT "(DISSE ORD SKAL EFTERFØLGES AF"
6190 PRINT "NAVNEORD - F.EKS: 'SPISE MAD'),"
6200 GOSUB 6590
6210 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
6215 PRINT "GEMME - FORTSÆTTE:"
6220 PRINT "VIRKER KUN, HVIS DE EFTER-"
6230 PRINT "FØLGES AF ORDET 'SPIL' (OG"
6240 PRINT "DET ER EN GOD IDE AT GEMME"
6250 PRINT "SPILLET F.EKS. FØR MAN PRØVER"
6260 PRINT "KRÆFTER MED EN UDFORDRING. SÅ"
6270 PRINT "KAN MAN NEMLIG BAGEFTER FORT-"
6280 PRINT "SÆTTE SPILLET, HVIS DET"
6290 PRINT "SKULLE GÅ GALT."
6300 GOSUB 6590
6310 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
6315 PRINT "STOP: STOPPER SPILLET."
6320 PRINT "HJÆLP: VED DU ALT OM."
6330 PRINT "SEJLE - FLYGTE: BRUGES UDEN NAVNEORD."
6340 PRINT "STATUS: VISER HVAD DU BÆRER."
6350 PRINT:PRINT "DIN BÆREEVNE OG DEN TID,"
6360 PRINT "DET TAGER DIG AT TILBAGELÆGGE"
6370 PRINT "STRÆKNINGEN MELLEM TO STEDER,"
6380 PRINT "ER AFHÆNGIG AF TINGENES VÆGT"
6390 PRINT "OG AF DINE KRÆFTER (DU KAN"
6400 PRINT "FA EKSTRA KRÆFTER VED AT"
6410 PRINT "SPISE ELLER SOVE. OG HVIS"

```

```

6420 PRINT "DU KAN FA NOGEN TIL AT HJÆLPE"
6430 PRINT "DIG MED AT BÆRE, ER DU FRI"
6440 PRINT "FOR DETTE PROBLEM)."
6450 GOSUB 6590
6460 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
6465 PRINT "ET SIDSTE LILLE TIP:"
6470 PRINT "DU KAN VED AT INDTASTE"
6480 PRINT "'R' PLUS 'RETURN' RYDDE SKÆRMEN"
6490 PRINT "OG SE, HVOR DU ER"
6500 PRINT "(HVIS URET OG STEDSANGIVELSEN"
6510 PRINT "FOROVEN ER FORSVUNDET),"
6520 PRINT "MEN DET KOSTER DIG"
6530 PRINT "ET MINUT HVER GANG."
6540 PRINT:PRINT "OG HUSK:DU SKAL VÆRE FÆRDIG"
6550 PRINT "INDEN D. 10. JULI KL. 12!"
6560 GOSUB 6590
6570 GOTO 2080
6589 REM MELLEMRUMSTANGENTROUTINE *****
6590 PRINT:PRINT "TRYK PÅ MELLEMRUMSTANGENTEN,"
6600 PRINT "NAR DU VIL FORTSÆTTE."
6610 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
      GET A$ *****
6615 IF A$="" THEN 6610
6620 IF A$(">)" THEN 6610
6630 RETURN
6649 REM STATUS *****
6650 PRINT R$
6660 BP=0
6670 FOR I=1 TO GL:IF G(I)=-1 THEN PRINT G$(I):BP=1
6680 NEXT I
6690 IF BP=0 THEN PRINT "INGENTING"
6700 GOTO 2450
6719 REM GEMME *****

```

```

6720 IF NO=33 THEN 6750:REM HAR DU KASSETTESTATION,
      SA SE FORKLARING TIL SAMME STED I "DET MYSTISKE
      RUMSKIB" *****
6730 PRINT "DET ENESTE, DU KAN GEMME,":PRINT "ER
      SPILLET.":GOTO 2450
6740 REM HAR DU KASSETTESTATION, SA SE FORKLARING
      TIL SAMME STED I "DET MYSTISKE RUMSKIB" *****
6750 OPEN "0",1,"AM-DATA":REM COMMODORE-BRUGER,
      SKRIV I STEDET: OPEN 1,1,1,"AM-DATA" *****
6760 PRINT# 1,BA
6770 PRINT# 1,TY
6780 PRINT# 1,IN
6790 PRINT# 1,TV
6800 PRINT# 1,HU
6810 PRINT# 1,LM
6820 PRINT# 1,RE
6830 PRINT# 1,TR
6840 PRINT# 1,SP
6850 PRINT# 1,BJ
6860 PRINT# 1,BF
6870 PRINT# 1,VF
6880 PRINT# 1,LO
6890 PRINT# 1,BE
6900 PRINT# 1,MI
6910 PRINT# 1,DA
6920 PRINT# 1,BD
6930 PRINT# 1,LE
6940 PRINT# 1,AZ
6950 PRINT# 1,TS
6960 PRINT# 1,NA$
6970 FOR I=1 TO GL:PRINT# 1,G(I):NEXT I
6980 FOR I=1 TO GL:PRINT# 1,G$(I):NEXT I
6990 FOR I=1 TO AU:PRINT# 1,U$(I):NEXT I
7000 FOR I=1 TO AN:PRINT# 1,N$(I):NEXT I

```

```

7010 FOR I=1 TO 4:PRINT# 1,V$(I):NEXT I
7020 FOR I=1 TO 4:FOR J=1 TO 8:PRINT# 1,L(I,J):NEXT
      J:NEXT I
7030 CLOSE 1
7040 CLS:PRINT "ØNSKER DU AT FORTSÆTTE SPILLET?"
7050 GOSUB 9270
7060 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
      GET A$ *****
7065 IF A$="" THEN 7060
7070 IF A$="J" THEN 2080
7080 IF A$="N" THEN END
7090 GOTO 7060
7109 REM FORTSÆTTE *****
7110 IF NO=33 THEN 7140:REM HAR DU KASSETTESTATION,
      SA SE FORKLARING TIL DET SAMME STED I "DET
      MYSTISKE RUMSKIB" *****
7120 PRINT "DET ENESTE,":PRINT "DU KAN FORTSÆTTE, ER
      SPILLET.":GOTO 2450
7130 REM HAR DU KASSETTESTATION, SA SE FORKLARING
      TIL DET SAMME STED I "DET MYSTISKE RUMSKIB"
      *****
7140 OPEN "I",1,"AM-DATA":REM COMMODORE-BRUGER,
      SKRIV I STEDET: OPEN 1,1,0,"AM-DATA" *****
7150 INPUT# 1,BA
7160 INPUT# 1,TY
7170 INPUT# 1,IN
7180 INPUT# 1,TV
7190 INPUT# 1,HU
7200 INPUT# 1,LM
7210 INPUT# 1,RE
7220 INPUT# 1,TR
7230 INPUT# 1,SP
7240 INPUT# 1,BJ
7250 INPUT# 1,BF
7260 INPUT# 1,VF

```

```

7270 INPUT# 1,LO
7280 INPUT# 1,BE
7290 INPUT# 1,MI
7300 INPUT# 1,DA
7310 INPUT# 1,BD
7320 INPUT# 1,LE
7330 INPUT# 1,AZ
7340 INPUT# 1,TS
7350 INPUT# 1,NA$
7360 FOR I=1 TO GL:INPUT# 1,G(I):NEXT I
7370 FOR I=1 TO GL:INPUT# 1,G$(I):NEXT I
7380 FOR I=1 TO AU:INPUT# 1,U$(I):NEXT I
7390 FOR I=1 TO AN:INPUT# 1,N$(I):NEXT I
7400 FOR I=1 TO 4:INPUT# 1,V$(I):NEXT I
7410 FOR I=1 TO AL:FOR J=1 TO 8:INPUT# 1,L(I,J):NEXT
      J:NEXT I
7420 CLOSE 1:GOTO 2080
7439 REM ANTÆNDE *****
7440 IF G(17)=-1 THEN 7460
7450 PRINT L$;"T AT TÆNDE MED.":GOTO 2450
7460 IF NO<>11 THEN 7590
7470 IF G(11)<>LO AND G(11)<>-1 THEN PRINT "DEN ER
      IKKE HER.":GOTO 2450
7480 IF G(11)<>-1 THEN 7510
7490 PRINT "MAN SKAL LÆGGE DYNAMIT FRA SIG,":PRINT
      "FØR MAN ANTÆNDER DET."
7500 PRINT "DU BLIVER SLAET IHJEL.":PRINT "HVOR DUM
      KAN MAN VÆRE?":GOTO 9080
7510 IF LO=38 THEN 7560
7520 G(11)=0:PRINT "DYNAMITTEN EKSPLODERER."
7530 PRINT "NU HAR DU IKKE NOGET DYNAMIT.":PRINT "DU
      SKULLE HAVE FYRET DET AF"
7540 PRINT "ET ANDET STED, DIT KVAJ!":PRINT "ET GODT
      RAD: BEGYND FORFRA,"
7550 PRINT "HVIS DU VIL FINDE SKATTEN.":GOTO 2450

```

```

7560 TV=1:G(11)=0
7570 PRINT "DEN TILMUREDE INDGANG":PRINT "RYGER I
      LUFTEN,"
7580 PRINT "OG DU KAN GA IGennem DEN.":N=2:GOSUB
      9240:GOTO 2080
7590 PRINT G$:GOTO 2450
7609 REM TØMME *****
7610 IF NO<>14 THEN 7730
7620 IF G(14)<>-1 AND G(14)<>LO THEN PRINT "DU HAR
      DEN IKKE.":GOTO 2450
7630 IF VF=1 THEN PRINT E$:GOTO 2450
7640 IF LO<>42 OR HU=1 THEN 7660
7650 GOTO 7690
7660 PRINT "DU HÆLDER FLASKENS":PRINT "INDHOLD UD PA
      JORDEN."
7670 PRINT "DET VAR IKKE SÆRLIG SMART.":VF=1
7680 PRINT O$;"!":GOTO 2450
7690 PRINT "DU TØMMER FLASKEN NED I HULLET,"
7700 PRINT "OG EN SKJULT MEKANISME AKTIVERES."
7710 PRINT "ET STENPANEL GLIDER TIL SIDE":PRINT "OG
      ABENBARER EN LILLE NICHE."
7720 VF=1:HU=1:N=4:GOSUB 9240:GOTO 2080
7730 PRINT "VÆR NU LIDT ALVORLIG, HVA'?:GOTO 2450
7749 REM STOPPE *****
7750 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
7755 PRINT "DU ØNSKER AT STOPPE SPILLET?"
7760 GOSUB 9270
7770 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
      GET A$ *****
7775 IF A$="" THEN 7770
7780 IF A$="J" THEN 7810
7790 IF A$="N" THEN 2450
7800 GOTO 7750

```



```

7810 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
7815 PRINT "ØNSKER DU AT GEMME SPILLET?"
7820 GOSUB 9270
7830 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
      GET A$ *****
7835 IF A$="" THEN 7830
7840 IF A$="J" THEN 6740
7850 IF A$="N" THEN END
7860 GOTO 7810
7879 REM SKÆRMTEKSTRUTINE *****
7880 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
7885 GOSUB 8450:REM NED TIL UR *****
7890 IF (LO=1 OR (LO>2 AND LO<18) OR (LO>22 AND
      LO<26)) THEN 7910
7900 GOTO 7920
7910 IF MI>1200 AND G(1)=-1 THEN LM=3
7920 BX=INT(RND(1)*2)+2
7930 IF (LO=1 OR (LO>2 AND LO<18) OR (LO>22 AND
      LO<26)) THEN 7950
7940 GOTO 7960
7950 IF MI>1200 AND G(1)<>-1 THEN LM=BX
7960 IF LM=1 THEN PRINT:PRINT "DET ER BÆLGMØRKT.":
      GOTO 7980
7970 GOTO 7990
7980 PRINT "DU KAN IKKE SE EN PIND.":RETURN
7990 IF LM=2 AND LO<>1 THEN PRINT:PRINT "DET ER
      MØRKT OG":GOTO 8010
8000 GOTO 8020
8010 PRINT "DET ER OVERSKYET,":PRINT "OG DU KAN
      INTET SE.":RETURN
8020 IF LM=3 OR (LM=2 AND LO=1) THEN 8040
8030 GOTO 8050

```

```

8040 PRINT S$+"JUNGLEN LIGGER BADET"+S$+"I MANENS
      KLARE LYS."
8050 IF AZ<>3 THEN 8080
8060 PRINT:PRINT "DU HAR IKKE FAET":PRINT "NOGET AT
      SPISE I FEM DØGN."
8070 PRINT "DU ER STILLE GLEDET":PRINT "IND I DEN
      EVIGE HVILE.":GOTO 9080
8080 IF DA>9 AND MI>720 GOTO 8990
8090 IF G(15)=-1 AND LO=7 THEN 9030
8100 DU$="DU ER ALENE. DU "
8110 IF BA=1 THEN DU$="DU OG "+N$+" "
8120 PRINT:PRINT DU$+L$(LO)+AP$
8130 IF AZ<>1 THEN 8180
8140 PRINT "DET ER LÆNGE SIDEN,":PRINT "DU SIDST FIK
      NOGET AT SPISE."
8150 PRINT "FAR DU IKKE NOGET AT SPISE":PRINT "I
      LØBET AF ET PAR DAGE"
8160 PRINT "VIL DET GA DIG ILDE."
8170 BE=BE+2:AZ=2:SP=DA+2:N=3:GOSUB 9240:GOTO 7880
8180 IF AZ<>4 THEN 8220
8190 PRINT "DU ER MEGET AFKRÆFTET.":PRINT "HVIS DU
      IKKE FAR NOGET"
8200 PRINT "AT SPISE INDEN MIDNAT,":PRINT "DØR DU AF
      SULT."
8210 BE=BE+4:AZ=5:N=3:GOSUB 9240:GOTO 7880
8220 IF SO<>1 THEN 8250
8230 SO=2:MI=MI+480:PRINT "DU ER BLEVET TRÆT"
8240 PRINT "OG LÆGGER DIG TIL AT SOVE.":N=3:GOSUB
      9240:GOTO 7880
8250 IF SO<>2 THEN 8280
8260 PRINT "DU HAR NU SOVET I OTTE TIMER."
8270 SO=0:BE=BE-2:N=2:GOSUB 9240:GOTO 7880
8280 IF BA<>1 AND BE>8 THEN 8300
8290 GOTO 8330

```

```

8300 PRINT "DU ER TEMMELIG AFKRÆFTET,":PRINT "OG DU
      SLÆBER PÅ FOR MEGET."
8310 PRINT "MED ANDRE ORD:":PRINT "DU KOMMER IKKE
      VIDERE,"
8320 PRINT "FØR DU HAR LAGT ET ELLER ANDET.":GOSUB
      8840:RETURN
8330 FOR I=1 TO GL:IF G(I)<>LO THEN 8370
8340 PRINT "DU KAN SE: "
8350 FOR I=1 TO GL:IF G(I)=LO THEN PRINT G$(I)
8360 NEXT I:PRINT:GOTO 8380
8370 NEXT I
8380 GOSUB 8670
8390 PRINT "DU KAN GÅ:"
8400 FOR I=1 TO 4:IF L(LO,I)<>0 THEN PRINT V$(I)
8410 NEXT I
8420 IF LO=3 AND BD=0 THEN PRINT "ELLER SEJLE"
8430 RETURN
8449 REM UR *****
8450 A=MI/60
8460 TM=INT(A)
8470 MN=INT(MI)-(TM*60)
8480 MN$=STR$(MN)
8490 IF SP=35 AND MI>1080 THEN AZ=1
8500 IF SP=36 AND AZ=DA+8 AND MI>1080 THEN AZ=1
8510 IF SP<35 AND DA>(SP+1) THEN AZ=3:GOTO 8540
8520 IF AZ=5 THEN 8540
8530 IF SP<35 AND DA>SP THEN AZ=4
8540 IF SO=2 THEN 8590
8550 IF MI>1439 THEN 8570
8560 GOTO 8590
8570 MI=MI-1440:TM=TM-24:DA=DA+1
8580 SO=1
8590 TM$=STR$(TM)
8600 DA$=STR$(DA)

```

```

8610 PRINT "DATO:";DA$;" . JULI * ";
8620 A$=RIGHT$(MN$,2)
8630 IF LEN(MN$)=2 THEN A$="0"+RIGHT$(MN$,1)
8640 PRINT "KLOKKEN ER";TM$;" : ";A$
8650 RETURN
8669 REM TRUSLER ETC *****
8670 IF BF<>LO THEN 8690
8680 PRINT "HER MØDER DU":PRINT "DINE BÆRERE
      IGEN.":BA=1:BF=0:PRINT
8690 Q$="HER BLIVER DU ANGREBET"+S$+"AF EN"
8700 IF LE=LO THEN PRINT Q$;" KÆMPESTOR LEOPARD.":
      PRINT
8710 IF TS=1 AND TR=LO AND G(15)=-1 THEN 8730
8720 GOTO 8770
8730 PRINT "HER BLIVER DU OVERRASKET":PRINT "AF EN
      DRILAGTIG ABE,"
8740 PRINT "DER SNUPPER SKATTEN FRA DIG,":PRINT
      "STIKKER AF OG SMIDER DEN"

```



```

8750 PRINT "GUDERNE MA VIDE HVOR.":PRINT
8760 G(15)=INT(RND(1)*17)+26:IF G(15)=LO THEN 8760
8770 IF LO=6 AND BJ=0 THEN PRINT Q$;" KEMPESTOR
      BJØRN.":PRINT
8780 IF TS=2 THEN TR=INT(RND(1)*5)+11:TS=3
8790 IF (TS=2 OR TS=3) AND TR=LO THEN 8810
8800 GOTO 8820
8810 PRINT Q$;" BLODTØRSTIG PANTER.":PRINT
8820 RETURN
8839 REM TVANGSLÆGGERUTINE *****
8840 PRINT:INPUT "HVAD VIL DU LÆGGE";A$
8850 NO=0
8860 IF A$="STATUS" THEN PRINT:PRINT R$:GOTO 8880
8870 GOTO 8900
8880 FOR I=1 TO GL:IF G(I)=-1 THEN PRINT G$(I)
8890 NEXT I:GOTO 8840
8900 FOR I=1 TO GL:IF LEFT$(A$,3)=N$(I) THEN NO=I
8910 NEXT I
8920 IF G(NO)<>-1 THEN 8840
8930 G(NO)=LO
8940 IF NO=3 OR NO=13 THEN BE=BE-5
8950 IF NO<>3 AND NO<>13 THEN BE=BE-1
8960 PRINT:PRINT "UDMÆRKET."
8970 RETURN
8989 REM AFSLUTNINGSRUTINE *****
8990 PRINT:PRINT "SOM DU KAN SE, HAR DU":PRINT
      "OVERSKREDET TIDSFRISTEN."
9000 PRINT "DU HØRER EN BULDREN AF VAND,":PRINT "DER
      KOMMER BRUSENDE IMOD DIG."
9010 PRINT "BEKLAGER, ";NA$:PRINT "DU NAEDE DET IKKE
      DENNE GANG,"
9020 PRINT "MEN DU ER VELKOMMEN":PRINT "TIL AT AT
      PRØVE IGEN.":GOTO 9090
9030 PRINT "BRAVO, ";NA$;"!":PRINT "DU ER NAET
      TILBAGE TIL"

```

```

9040 PRINT "DIN LEJR MED SKATTEN."
9050 PRINT "DU HAR GENNEMFØRT SPILLET."
9060 PRINT "TIL LYKKE!":GOTO 9090
9079 REM DØDSROUTINE *****
9080 PRINT:PRINT "BEKLAGER, ";NA$:PRINT "...SADAN ER
LIVET."
9090 PRINT:PRINT "HAR DU LYST TIL":PRINT "AT PRØVE
EN GANG TIL?"
9100 GOSUB 9270
9110 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
GET A$ *****
9115 IF A$="" THEN 9110
9120 IF A$="J" THEN 9150
9130 IF A$="N" THEN END
9140 GOTO 9090
9150 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
CHR$(147) *****
9155 PRINT "HVIS DU HAR GEMT SPILLET":PRINT "OG
ØNSKER AT FORTSÆTTE,"
9160 PRINT "SA TRYK PÅ 'F' FOR 'FORTSÆT'."
9170 PRINT "HVIS DU VIL BEGYNDE HELT FORFRA,"
9180 PRINT "SA TRYK PÅ 'B' FOR":PRINT "'BEDRE HELD
NÆSTE GANG'."
9190 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
GET A$ *****
9195 IF A$="" THEN 9190
9200 IF A$="F" THEN 7130
9210 IF A$="B" THEN RUN
9220 GOTO 9190
9239 REM LØKKERROUTINE *****
9240 FOR I=1 TO (N*1000):NEXT I
9250 RETURN
9269 REM JA/NEJ-ROUTINE *****
9270 PRINT:PRINT "TRYK PÅ 'J' FOR 'JA':PRINT "ELLER
'N' FOR 'NEJ'."

```

9280 RETURN

9299 REM OM IGEN-RUTINE *****

9300 PRINT "PRØV NOGET ANDET, ";NA\$;"!":GOTO 2450

Variablerne i spillet

Inden gennemgangen af spillet kommer her en liste over de variabler, der er brugt i det. Jeg har ikke taget sættet G() med, da det består af tal, der hele tiden ændrer sig (når du tager tingene med dig eller lægger dem et andet sted). Begyndelsestallene for G() kan du se i linierne 930 og 2020-2060. Se også forklaringen til disse linier.

Det ville også være spild af plads at tage sættet L\$, som indeholder lokalitetsbeskrivelserne, med her. Dem kan du se i LIST-ningen fra linie 1060 og fremefter. Og sættet V\$, der kun indeholder de fire verdenshjørner, kender du også i forvejen.

Heller ikke sættet L(), der indeholder DATA for verdenshjørner og minutter, har jeg medtaget. Du kan se de enkelte elementer i sættet fra linie 1090 og fremefter (DATA-sætningerne), hvor de står under de lokalitetsbeskrivelser, de hører til. Husk, at de første fire tal i hver DATA-sætning er verdenshjørner, og at de sidste fire er de minutangivelser, der hører til disse verdenshjørner.

De fleste af strengvariablerne er heller ikke med i denne oversigt. Du kan se en hel del af dem i linierne 660-840, og hvis du støder på andre end disse, er det lokale variabler, der kun bliver brugt midlertidigt over ganske få linier, og hvis betydning burde fremgå tydeligt af selve programmet omkring disse få linier. Der er dog to undtagelser, og de kommer her:

AP\$: Appendiks til lokalitetsbeskrivelsen. Normalt er AP\$ lig med CHR\$(13) (linieskift).

NA\$: Den strengvariabel, der indeholder spillerens navn, som maskinen i begyndelsen af spillet beder om at få oplyst.

Indekserede variabler:

G\$(1): EN LOMMELYGTE
G\$(2): NOGET PROVIANT
G\$(3): NOGET AMMUNITION
G\$(4): EN SØLVNØGLE
G\$(5): EN KNIV
G\$(6): EN MACHETE
G\$(7): EN DÅSEÅBNER
G\$(8): EN DAGBOG
G\$(9): EN GULDNØGLE
G\$(10): EN RIFFEL
G\$(11): NOGET DYNAMIT
G\$(12): ET TOV
G\$(13): EN JERNNØGLE
G\$(14): EN FLASKE
G\$(15): EN SMARAGDKRONE
G\$(16): ET SKRIN
G\$(17): NOGLE TÆNDSTIKKER
G\$(18): EN DØD LEOPARD
G\$(19): EN DØD BJØRN

U\$(1): GÅ
U\$(2): SKYde
U\$(3): ÅBNe
U\$(4): TAGe
U\$(5): LÆGge
U\$(6): BRUge
U\$(7): KØBe
U\$(8): DRIkke
U\$(9): UNDersøge
U\$(10): SPIse
U\$(11): LÆSe
U\$(12): GEMme
U\$(13): FORtsætte
U\$(14): ANTænde
U\$(15): TØMme

OBS! Husk at værdien af UO svarer til indholdet af U\$(). Hvis der i programmet står at UO=5, betyder det, at det inputtede ud-sagnsord er U\$(5), altså »LÆGge«.

- N\$(1): LOMmelygte
- N\$(2): PROviant
- N\$(3): AMMunition
- N\$(4): SØLvnøgle
- N\$(5): KNIv
- N\$(6): MACHete
- N\$(7): DÅSeåbner
- N\$(8): DAGbog
- N\$(9): GULdnøgle
- N\$(10): RIFFel
- N\$(11): DYNamit
- N\$(12): TOV
- N\$(13): JERNnøgle
- N\$(14): FLAske
- N\$(15): SMARagdkrone
- N\$(16): SKRin
- N\$(17): TÆNDstikker
- N\$(18): LEOpard
- N\$(19): BJØrn
- N\$(20): INDFødte
- N\$(21): PANter
- N\$(22): TRUssel
- N\$(23): DØR
- N\$(24): BÅD
- N\$(25): NØGLE
- N\$(26): VAND
- N\$(27): WHIsky
- N\$(28): TYKning
- N\$(29): RUIn
- N\$(30): BRO
- N\$(31): MAD
- N\$(32): ETIkette
- N\$(33): SPIl

N\$(34): NORdpå
N\$(35): SYDpå
N\$(36): ØSTpå
N\$(37): VESTpå



OBS! Husk at værdien af NO svarer til indholdet af N\$(). Hvis der i programmet står at NO=7, betyder det, at det inputtede navneord er N\$(7), altså »DÅSeåbner«. Bemærk dog, at der pilles lidt ved denne regel i gå-rutinen og tvangslægge-rutinen.

Enkeltværdivariables

Disse er delt op i globale og lokale variable. Globale variable optræder hele spillet igennem, og det er dem, man er nødt til at GEMme (foruden de fleste af de indekserede), hvis man senere vil kunne vende tilbage til det samme sted i spillet. De lokale variable er blot til hjælp undervejs og har ingen gennemgående betydning.

Globale variable

AL: Antallet af lokaliteter.

AN: Antallet af navneord.

AU: Antallet af udsagnsord

AZ: Variabel, der ligesom SP bruges i beregningen af, hvor længe siden det er, du har fået noget at spise.

AZ=1: Der er gået mindst 18 timer, siden du sidst fik noget at spise. Du får første advarsel.

AZ=4: Der er gået fire døgn, siden du sidst fik noget at spise. Du får sidste advarsel.

AZ=3: Der er gået fem døgn. Du dør af sult.

BA: Bærere.

BA=0: Du har ingen bærere.

BA=1: Du har bærere.

BA=2: Bærerne er stukket af nede i hulen.

BD: Båd.

BD=0: Du har ikke brugt den.

BD=1: Du har brugt den.

BE: Bæreevne.

Hvis $BE > 8$ kan du ikke bære på mere (medmindre du har bærerne med).

BF: Bærere flygtet/ikke flygtet.

BF=0: De er ikke flygtet.

BF=A: De er flygtet og befinder sig på Lokalitet nr. A.

BJ: Bjørnen.

BJ=0: Den er levende.

BJ=1: Den er død.

DA: Dato.

GL: Antallet af genstande, der befinder sig på en lokalitet.

HU: Panelet i skattehulen.

HU=0: Det er lukket.

HU=1: Det er åbent.

IN: Indgang til skattehulen.

IN=0: Den er lukket.

IN=1: Den er åben.

LE: Leopards lokalitet.

LE=0: Leoparden er død.

LE=A: Leoparden befinder sig på Lokalitet nr. A.

LM: Lyst/mørkt.

LM=0: Det er lyst.

LM=1: Det er mørkt inde i en hule.

LM=2: Det er mørkt og overskyet udenfor.

LM=3: Det er mørkt udenfor, men månen er fremme.

LO: Lokalitet.

MI: Minutter.

RE: Reb (Når jeg ikke bruger variabelen TO – for TOv – skyldes det, at TO er et af BASICs reserverede ord).

RE=0: Intet tov på Lokalitet nr. 32 eller 35.

RE=1: Tov på Lokalitet nr. 32 eller 35.

RE=2: Tovet er taget fra Lokalitet nr. 35. Du kan ikke komme ud af hulen igen.

SP: Variabel, der ligesom AZ bruges i beregningen af, hvor længe siden det er, du sidst har fået noget at spise.

SP=35: Denne værdi har SP ved spillets begyndelse.

SP=36: Du har lige spist.

SP<35: Der er gået mindst 18 timer, siden du sidst fik noget at spise.

TR: Trussels lokalitet.

TS: Trussels art

TS=1: Truslen er åben.

TS=2: Truslen er panteren (og den skifter lokalitet).

TS=3: Truslen er panteren (og den bliver på stedet).

TV: Tilmuret væg.
TV=0: Den er hel.
TV=1: Den er sprængt.

TY: Tykning.
TY=0: Den er ufarbar.
TY=1: Den er farbar.

VF: Flasken.
VF=0: Den er fuld.
VF=1: Den er tom.

Lokale variabler

AY, AX: Bruges til at bestemme, hvilket sted, du havner, når du flygter.

BP: Bruges i forbindelse med opgørelse af status.

BX: Bruges til beregning af, om månen skal være skjult eller fremme.

FL: Bruges til beregning af, om du slipper væk eller bliver dræbt, når du forsøger at flygte.

KU: Bruges til beregning af, om du rammer leoparden eller ej, når du skyder efter den.

N: Vil du se ret tit. En pladsbesparende variabel, der altid bliver brugt i forbindelse med et GOSUB 9240. I linie 9240 bliver N ganget med 1000. Teksten bliver stående på skærmen i den tid, det tager maskinen at gennemløbe pauseløkken FOR I=1 TO (N*1000):NEXT I. Det sker de steder, hvor man ikke vil kunne nå at læse teksten, dersom denne pause ikke er med.

SE: Bruges til beregning af, om du bliver dræbt eller overlever, når du sejler ned ad vandfaldet.

SO: Bruges til beregning af, hvornår du skal sove.

TM, MN, MN\$, TM\$, DAS: Bruges i den rutine, der ender med at PRINTe spilleuret ud på skærmen.

Gennemgang af Amazon-spillet

I denne gennemgang af spillet koncentrerer jeg mig om ting, jeg ikke før har uddybet fyldestgørende. Hvis du savner en forklaring til et eller andet, så burde ovenstående variabeliste kunne give dig et fingerpeg. Ellers prøv at kigge på gennemgangen af demonstrationsspillet eller »Det mystiske rumskib«.

80-90: Maskinen beder om dit navn for at kunne bruge det nu og da, når den »snakker« til dig. Jeg synes selv, det virker hyggeligt, men du er naturligvis fri til at udelade det, hvis du vil.

250-260: Her finder vi ved hjælp af maskinens indbyggede ur (og datoen) et grundtal til beregning af tilfældige tal. Hvis din maskine ikke kræver et sådant grundtal, så slet disse to linier. Hvis din maskine kræver et grundtal (»Random Number Seed«), men din BASIC ikke rummer ordene TIME\$ og DATE\$, så se i afsnittet »Tilfældigheder« i forrige kapitel, hvad du skal gøre.

270: Gå ned til mellemrumstangent-rutinen. Hvis du i forrige linie bruger en løkke til frembringelse af et tilfældigt grundtal (se afsnittet »Tilfældigheder« i forrige kapitel), skal du slette denne linie og i stedet gøre, præcis hvad der står i forklaringen (altså den i forrige kapitel).

640: StartLOkaliteten sættes til 7, MINutantallet til 720 (12 timer), DAToen til 1, og Trussels art (TS) til 1. TS=1 betyder, at aben i hulen er klar til at hugge kronen fra dig. LE=1 betyder, at leoparden ikke er død. 1-tallet kunne lige så godt være et andet tal. Det vigtigste er, at det ikke er nul, for så er leoparden død. Grunden til, at tallet er ligegyldigt, er at det alligevel bliver lavet om i linie

2100, inden spillet overhovedet kommer i gang. Men hvis tallet var nul, første gang maskinen kom til linie 2100, ville den registrere leoparden som død, og det skal den ikke være, før spillet har skudt den.

660-840: De mest brugte ord og sætninger forkortet til strengvariabler. Herved spares megen plads i resten af spillet. Bemærk, at vi sætter lineskift-koden CHR\$(13) (se afsnittet »Skærmpresentation« i forrige kapitel) til \$\$\$. Denne variabel kommer du til at se meget mere til.

860: Bemærk, at vi her giver det indekserede variabelsæt L() AL gange 8 elementer i modsætning til de AL gange 4 elementer, vi gav det i »Det mystiske rumskib«. Hvis du ikke forstår hvorfor, så kig under afsnittet »Tidsfrist« i forrige kapitel.

930: DATA for sættet G(), der bestemmer, på hvilke lokaliteter de forskellige genstande skal anbringes. De elementer, der her har værdien nul, anbringes først senere, enten tilfældigt (i linierne 2020-2060) eller hen ad vejen, efterhånden som forskellige betingelser bliver opfyldt.

940-980: Spillets genstande. Når de to sidste elementer i spillet er en *død* leopard og en *død* bjørn, skyldes det, at vi først kan tage dem med os, når de *er* døde – og før da bliver de ikke PRINTet ud på skærmen som »genstande«.

990-1000: Udsagnsordene. Nye er SKYde, ÅBNe, KØBe, DRiKke, SPiSe, ANTænde og TØMme. Hvis din maskine ikke har Æ, må du bruge AE, men så bliver LAEgge lig med LAEse (husk at computeren kun læser de første tre bogstaver). Dette problem kan du løse ved fx at skrive STUdere i stedet for LAEse, eller du kan give maskinen ordre til at læse de første *fire* bogstaver i stedet for de første tre. I så fald skal *alle* elementerne i dette sæt naturligvis være på fire bogstaver, og du skal lave 3-tallerne i linierne 2670 og 2680 om til 4-taller.

1060-1080: Da jeg har holdt mig til linier på højst 80 anslag, er

jeg nødt til at fordele indholdet af L\$(1), som er en *lang* lokalitetsbeskrivelse, over tre linier. Det gøres på denne måde. Hvis du er forvirret over, hvad der foregår her, så læs afsnittet »Skærmpresentation« i forrige kapitel. Bemærk, at variabelen S\$ står for CHR\$(13). Du kan se betydningen af denne og de andre strengvariabler, der er i spillets lokalitetsbeskrivelser, i linie 660-840.

1090: DATA for verdenshjørner og minutter til Lokalitet nr. 1. Se evt. afsnittet »Tidsfrist« i forrige kapitel.

1440-1490: Her har vi vores »snyde-labyrint«. Bemærk indholdet af de første fire elementer i DATA-sætningerne. Læs evt. den uddybende forklaring i afsnittet »Labyrinter« i forrige kapitel samt synopsis tidligere i dette kapitel.

2020-2060: Her fordeles nogle af spillets genstande tilfældigt rundt omkring i spillet. Bemærk, at vi i forvejen har været nødt til at placere dem med et nul i DATA-sætningen i linie 930 (for at sætte plads af til dem i sættet G()).

2080-2420: De ændringer, der finder sted undervejs i spillet, og som skal gennemlæses, hver gang computeren foretager sin »kontrolrunde«.

2080: AP\$ står for »Appendiks til lokalitetsbeskrivelse«. Hvis intet andet gælder, gøres AP\$ lig med et linieskift.

2090: Hvis LEoparden befinder sig på stedet, skal man ikke bare kunne fjerne den ved at rydde skærmen (se linie 2480). Derfor springes næste linie, der sender leoparden et andet sted hen, over.

2100: Hvis LE < > 0 (hvis leoparden ikke er død), anbringes leoparden tilfældigt et sted mellem LOKalitet nr. 1 og LOKalitet nr. 22. Men hvis stedet er 6 (hvor der i forvejen er en bjørn) eller 7 (det er lejren, og jeg syntes, det ville være lige lovlig barsk at risikere at starte spillet med at blive overfaldet af en leopard) eller

11 (man skal ikke kunne undslippe leoparden ved at hugge sig gennem tykningen – så hellere helt holde den væk) får maskinen besked på at finde et nyt tal.

2110: Hvis vi er i skattehulen og har smaragdkronen med, skal den kunne hugges af aben. Vi har godt nok allerede sat TS (Truslens art) til 1 (aben), men den kan jo blive ændret undervejs i spillet og skal have mulighed for at blive ændret tilbage igen, hvis forholdene kræver det.

2120-2130: Blot et lille trick, fordi det hele ikke kunne stå på linie 2140. Hvis din computer kan tage lange linier (fx en IBM PC), er der intet i vejen for, at du skriver linie 2120-2140 sammen til én linie, hvis du kan finde ud af det.

2140: Hvis man ikke har købt noget på handelsstationen, er det nok første gang, man er der, og så udskrives der et oplysende appendiks til lokalitetsbeskrivelsen.

2150: TS er det tal, der bestemmer truslens art, og hvis $TS=1$, er truslen en abe. Hvis $(15)=-1$, har man smaragdkronen med sig, og så risikerer man at løbe ind i den drilske abe, der stjæler fra en, på den lokalitet, hvor maskinen anbringer TR (står også for TRussel, men bestemmer, *hvor* truslen skal placeres, hvorimod TS bestemmer *hvad* den skal være).

2160-2170: Hvis man har smaragdkronen med sig og kommer tilbage til junglen, sættes TS (Truslens art) til 2 (en panter). Hvis man flygter fra panteren, smider man kronen fra sig, og så gøres TS til 3 (hvorved der sørges for, at panteren bliver på stedet, så man ikke bare kan gå tilbage og samle kronen op igen).

2180-2420: Resten af ændrings-rutinen burde du selv kunne tyde ved hjælp af den ovenstående variabelliste.

2450-2810: Strengdelings-rutinen. Du kender den i forvejen, men jeg har et par kommentarer:

2480: Hvis man rydder skærmen, får man et »strafpoint« på ét minut.

2490-2500: Kommandoen »sove« er ikke med i spillet, men der skal såmænd nok være en eller anden Karl Smart, der prøver den alligevel.

2560-2590: Herfra videresendes spillets enkeltordskommandoer i modsætning til dobbeltordskommandoerne, der skal deles. Bemærk, at »flygte« og »sejle« er nye kommandoer, som ikke var med i »Det mystiske Rumskib«.

2700: Hvis nogen prøver at dræbe noget, får de at vide, at de må prøve noget andet. Maskinen accepterer kun kommandoen »SKYde«, hvis der skal slås noget eller nogen ihjel.

2770-2790: Her tages højde for, at spilleren skriver »bærere« i stedet for »indfødte«, »hængebro« i stedet for »bro« og »skat« eller »krone« i stedet for »smaragdkrone«.

2880-2960: Nu er vi i gå-rutinen. Hvis leoparden, bjørnen eller en anden trussel er på stedet, kan man ikke gå nogen steder, men bliver nødt til at uskadeliggøre truslen eller flygte.

2970-2980: Her frasorterer vi alle andre »navneord« end de fire verdenshjørner og omnummererer dem til 1,2,3 og 4.

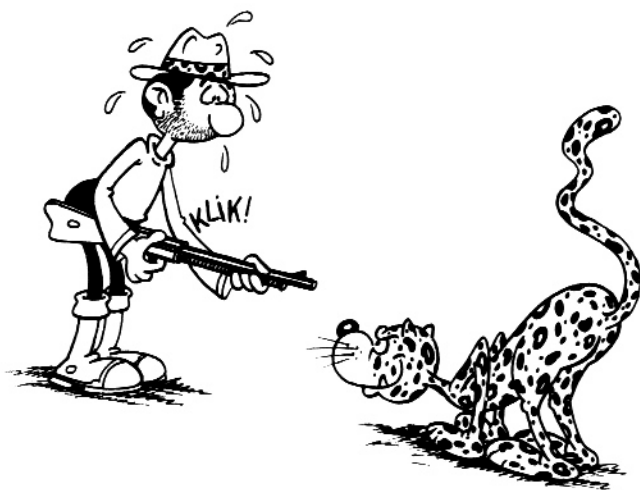
2990-3020: Hvis man står på afgrundens rand og alligevel går mod syd, styrter man ned i den og slår sig ihjel.

3040: Her ser du, hvordan vi regner ud, hvor mange minutter der skal gå pr. lokalitetsskift. Vi lader dem vokse, hvis der bæres på noget.

3070-3300: Flygte-rutinen. Hvis man prøver at flygte uden grund, ryster maskinen på hovedet (linie 3090), men ellers vil man i to tredjedele af tilfældene flygte og i én tredjedel af tilfæl-

dene blive slået ihjel. Hvis man overlever, smider man alt, hvad man bærer på og bliver anbragt på en tilgrænsende lokalitet. Der lægges kun det halve antal minutter til, fordi man løber *hurtigt*. Variablerne AY og AX er nødvendige ved beregningen af den anden lokalitet. Der ville gå kludder i det hele, hvis vi ikke (midlertidigt) brugte dem i stedet for LO og NO. Prøv bare at lade være og se, hvad der sker!

3320-3610: Skyde-rutinen. Har man hverken ammunition eller riffel, kan man ikke skyde noget, og hvis man skyder efter ingenting, får man også det at vide. Liniernes 3430-3440 sørger for, at man kun rammer leoparden én gang ud af fire. Rammer man ved siden af, flygter leoparden til et andet sted (linie 3470). Linie 3480 sørger for, at det virkelig *bliver* et andet sted. Og hvis man forsøger at skyde sine bærere (der skal nok være en eller anden klovn, der prøver det), vil det gå en ilde.



3630-3860: Åbne-rutinen. Skulle være nem at tyde, hvis du bruger variabellisten.

3880-4110: Tage-rutinen. Hvis du forsøger at tage båden med på nakken, får du et kvikt svar. Og hvis du forsøger at tage tovet på

hængebroen, får du besked på at undersøge, om det er der. Det dukker nemlig først op, når broen er blevet undersøgt, for førstegangsspillere skal ikke bare kunne se det. Det skal jo heller ikke være alt for nemt, vel? Og hvis du forsøger at tage noget på handelsstationen, får du at vide, at du pænt må betale for det, og det er jo egentlig også meget rimeligt. Hvis du bærer på for meget, får du også det at vide, men ellers får du lov til at tage det, du kan bære. Tingene i spillet har vægtværdien 1 undtagen ammunitionen, der vejer 5 gange så meget, og dynamitten, provianten, den døde bjørn og den døde leopard, som du slet ikke kan bære uden hjælp. Hvis du har bærere med, er der overhovedet ingen problemer. Hvis du er på Lokalitet nr. 35 og tager tovet, dummer du dig (linie 4090).

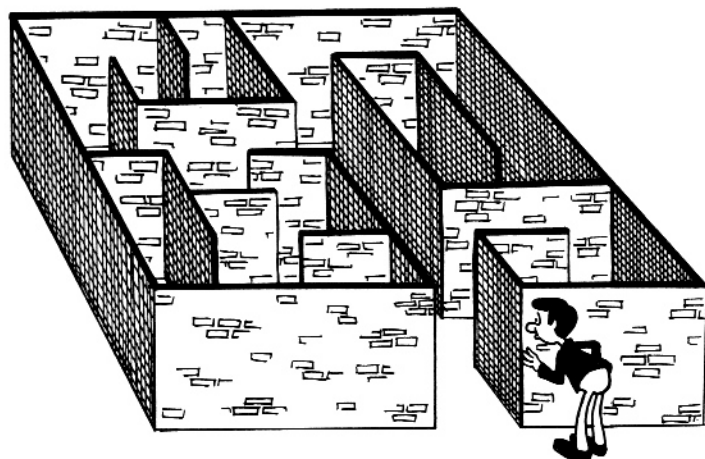
4130-9300: Og så videre og så videre. Resten af spillet kører efter principper, der allerede er gennemgået, og du burde ud fra gennemgangen af demonstrationsspillet, »Det mystiske rumskib«, »Nogle videre finesser«, variabellisten og gennemgangen af den første del af dette spil kunne finde ud af at tyde resten selv. Et par sidste ting vil jeg dog lige nævne: Når vi GEMmer SPillet, tager vi ikke lokalitetsbeskrivelsessættet med, da maskinen nægter at tage lineskift-instruktionen CHR\$(13) med i en fil – ja, faktisk kun tager det, der står *inden* et sådant CHR\$(13) med – men det er også ligegyldigt, da de ændringer, der eventuelt har fundet sted i lokalitetsbeskrivelserne, vil blive læst af maskinen, når vi FORTsætter SPillet og den tager sin kontrolrunde (og det gør den, hver gang den får et input efter at have spurgt HVAD VIL DU GØRE?).

Noget nyt i dette spil er også tvangslægge-rutinen (linierne 8840-8970), som træder i kraft, når det er et stykke tid siden, man har fået noget at spise, og har mistet kræfter. Maskinen tager kræfter fra en ved at lægge tal til ens bærevne, og når bærevnen (BE) kommer op over 8, vil man blive bedt om at lægge et eller andet, indtil den er acceptabel igen. Man kan faktisk ikke komme videre, før man har gjort det. Jeg har i denne rutine indført mulighed for at bede om en STATUS (da det er en subrutine, som man ikke må bryde ud af, før maskinen når et RE-

TURN, nytter det ikke noget at hoppe op til status-rutinen midt i det hele). Jeg sætter også i begyndelsen af rutinen NO til 0, da jeg har brug for et NO i de følgende linier og maskinen stadig har et gammelt NO stående i sin hukommelse.

Og nu over til noget helt andet.

6 · Tilfældigt anbragte lokaliteter



Som vi så i »På skattejagt i Amazonlandet«, kunne vi skabe en vis variation i spillet ved at bruge tilfældighedstalsgeneratoren. Leoparden dukkede op på tilfældige steder, og mange af genstandene blev tilfældigt anbragt nye steder, hver gang man startede et spil.

Vi har altså mulighed for at anbringe vores ting tilfældigt, men den helt store variation giver dette naturligvis ikke, da vi stadig er nødt til at beholde lokaliteterne de steder, vi har anbragt dem. De hænger sammen efter et nøje tilrettelagt mønster, og dem kan vi ikke flytte rundt på . . . eller kan vi?

Ja, det kan vi faktisk.

Men det kræver, at vi griber kortlægningen lidt mere matematisk an, end vi hidtil har gjort. Prøv endnu en gang at se på kortet over Amazon-spillet. Det er der ikke mange muligheder for at lave om på. Det skal se ud, som det gør, og der ville ikke være nogen idé i at flytte tilfældigt rundt på lokaliteterne, da de

er stykket hensigtsmæssigt sammen og skal udforskes i en bestemt rækkefølge, for at spillet i sidste ende kan gå op. Der er kun én ulempe ved dette system: jo flere gange vi spiller spillet, jo bedre lærer vi det at kende, og til sidst gider vi ikke mere, da vi kender »stederne« ud og ind og lige præcis ved, hvordan vi skal løse det hele. Spillet er ikke længere en spændende kryds-ogtværs, men en kedelig udfyldningsopgave.

Nu er der ikke ret meget her i livet, der ikke bliver kedeligt af at blive gentaget for meget. Selv en spændende og velskrevet bog kan blive ret uinteressant, når man har læst den tredive gange. Men hvis vi fjernede sidetallene på den, klippede siderne ud og limede dem tilfældigt sammen, ville vi i hvert fald kunne få lidt tid til at gå med at hitte rundt i handlingen. Men så ville det ikke rigtig længere være en bog – snarere et puslespil.

Men kunne vi konstruere en bog, hvis sider blev tilfældigt anbragt hver gang, *uden* at det skabte forvirring i handlingen?

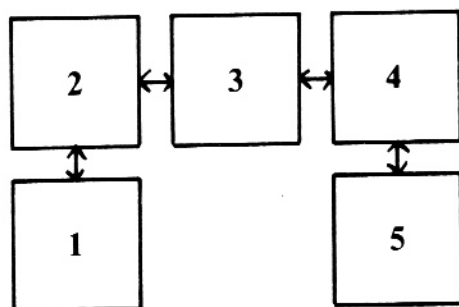
Næppe. Kunne vi konstruere et eventyrspil, hvor dette lod sig gøre? Ja, på en måde.

I et eventyrspil er der ikke »sider«, men der er lokaliteter, og der er genstande (»levende« eller »døde«). Vi ved, at vi kan anbringe genstandene tilfældigt. Lad os nu undersøge, hvordan vi gør det samme med lokaliteterne.

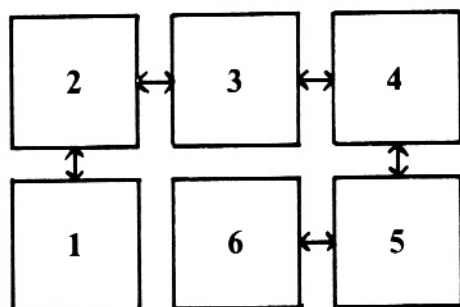
Allerede fra begyndelsen må vi gøre os én ting klart: Hvis vi anbringer lokaliteterne tilfældigt, kan vi ikke på forhånd vide, hvordan de kommer til at hænge sammen i forhold til hinanden – og det må vi så tilpasse vores forhindringer efter. Hvis alt er skruet tilfældigt sammen, er fx låste døre ret svære at administrere. Det nytter ikke meget, at vi anbringer en låst dør mellem Lokalitet nr. 24 og Lokalitet nr. 25, hvis vi ikke er hundrede procent sikre på, at man ikke kan gå uden om den ad en anden vej. Hvis vi ved, at Lokalitet nr. 25 ender *blindt*, kan vi naturligvis anbringe en låst dør og lægge et eller andet derind, som man kun kan få fat i, når man har fundet nøglen – det er klart. Men du kan sikkert tydeligt se, at denne type spil nødvendigvis må have sine begrænsninger på dette punkt. Til gengæld har det den fordel, at man får et helt nyt landskab at udforske, hver gang man spiller spillet.

Tilbage til kortlægningen. Hvis vi skal have maskinen til at

lave et tilfældigt kort, kræver det, at vi kan give den nøje besked om, at rummene skal hænge logisk sammen. Hvis vi kan gå mod syd, skal vi også kunne gå tilbage mod nord. Det er sådan set nemt nok at få maskinen til at lave en række rum, der hænger sammen på denne måde.



På ovenstående kort kan du se 5 rum, der er blevet tilfældigt anbragt efter denne recept. Vi har givet maskinen ordre til at anbringe Rum nr. 2 ved siden af Rum nr. 1 i en tilfældig retning, som her er blevet nord. Vi har givet den ordre til at registrere, at eftersom man kan gå mod nord fra Rum nr. 1, skal man også kunne gå mod syd fra Rum nr. 2. Det kan den sagtens finde ud af. Den kan også godt finde ud af at anbringe Rum nr. 3 tilfældigt ved siden af Rum nr. 2 og Rum nr. 4 tilfældigt ved siden af Rum nr. 3 og Rum nr. 5 tilfældigt ved siden af Rum nr. 4. Men når den vender tilbage til et sted, hvor den har været, bliver der problemer:



Nu har maskinen kortlagt Rum nr. 6, der tilfældigt er blevet anbragt ved siden af Rum nr. 5, og hvis den så tilfældigvis anbringer Rum nr. 7 vest for Rum nr. 6, kommer Rum nr. 7 til at falde sammen med Rum nr. 1, og logikken begynder at halte. For hvis Rum nr. 7 er lig med Rum nr. 1, bør man kunne gå mod nord fra Rum nr. 7, men det kan maskinen ikke vide noget om, da den betragter dem som to forskellige rum.

Vi må med andre ord finde ud af at fortælle maskinen, *hvor* Rum nr. 1 ligger, så den ikke begår denne fejl. Og hvordan gør vi så det?

Det gør vi ved på forhånd at placere rummene på *helt bestemte steder* på vores kort. Det er altså lidt af en tilsnigelse at sige, at *lokaliteterne* bliver tilfældigt anbragt i denne type spil. Som vi lige har set, lader dette sig strengt taget ikke gøre. Men vi kan få det til at *se ud*, som om det er tilfældet, ved at anbringe *dørene* tilfældigt.

Men før vi gør dette, er vi altså nødt til at fortælle maskinen, hvor rummene ligger i forhold til hinanden. Lad os kigge på det næste kort:

1	2	3
4	5	6
7	8	9

Vi har her et kort med ni rum anbragt fortløbende efter hinanden på tre »linier«. Et kort, der er bygget op på denne måde, vil

maskinen straks føle sig hjemme i. Den ved, hvor rummene ligger i forhold til hinanden (for det har vi fortalt den i programmet), og den vil så også have mulighed for at se, om den har været på et bestemt sted før. Den roder ikke længere rundt i blinde.

Lad os udvide dette kort til at omfatte 100 rum:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Disse 100 rum har vi anbragt fortløbende på 10 »linier« à 10 rum. Lad os sætte plads af i hukommelsen til de døre, der skal være i disse 100 rum:

10 DIM L(100,4)

Derefter giver vi maskinen besked på tilfældigt at give dørene en værdi, der enten skal være nul eller et positivt tal. Hvis døren er nul, kan man som bekendt ikke gå igennem den. Hvis vi får mange nuller, får vi ikke ret mange døre. Så lad os for eksperimentets skyld sige, at vi vil have i gennemsnit tre døre for hver tom væg i spillet:

```
20 FOR I=1 TO 100:FOR J=1 TO 4:L(I,J)=INT(RND(1)*4):  
    NEXT J:NEXT I
```

Herefter må vi straks sørge for at fortælle maskinen, at *ydevæggene* ikke må have døre. Hvis vi ikke gjorde dette og fx kunne gå mod nord fra Rum nr. 3, ville vi havne i Rum nr. -7, som ikke eksisterer i vores system, og som ikke er med i det DIMensionerede sæt. Maskinen ville protestere og afbryde programmet. Vi er altså nødt til at fortælle den, at ydevæggene i vores system ikke har nogen døre:

```
30 FOR I=1 TO 10:L(I,1)=0:NEXT I  
40 FOR I=10 TO 100 STEP 10:L(I,3)=0:NEXT I  
50 FOR I=91 TO 100:L(I,2)=0:NEXT I  
60 FOR I=1 TO 91 STEP 10:L(I,4)=0:NEXT I
```

Og når det er sket, må vi sørge for, at der bliver sammenhæng imellem dørene. At man, dersom man ikke kan gå mod syd fra fx Rum nr. 32, heller ikke skal kunne gå mod nord fra Rum nr. 42 osv. – og at man, dersom man kan gå mod øst fra fx Rum nr. 64, også skal kunne gå mod vest fra Rum nr. 65. Det gør vi på følgende vis:

```
70 FOR I=1 TO 100  
80 IF I<11 THEN 100  
90 IF L(I,1)=0 THEN L(I-10,2)=0
```

```

100 IF L(I,1)<>0 THEN L(I,1)=I-10
110 IF I>90 THEN 130
120 IF L(I,2)=0 THEN L(I+10,1)=0
130 IF L(I,2)<>0 THEN THEN L(I,2)=I+10
140 IF I=100 THEN 160
150 IF L(I,3)=0 THEN L(I+1,4)=0
160 IF L(I,3)<>0 THEN L(I,3)=I+1
170 IF I=1 THEN 190
180 IF L(I,4)=0 THEN L(I-1,3)=0
190 IF L(I,4)<>0 THEN L(I,4)=I-1
200 NEXT I

```

Et kort konstrueret med dette program kan fx komme til at se sådan ud:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Som vi kan se, er der ikke alt for god sammenhæng mellem rummene. Der er en hel del »øer«, som ikke er forbundet med resten af kortet, og det var jo vores hensigt at få det hele til at hænge sammen. Hvis vi prøver at forhøje 4-tallet i RND-sætningen i linie 20 til et 5-tal, ser vi, at hyppigheden af »øer« er formindsket, men til gengæld er kortet begyndt at hænge *lidt* for godt sammen. Det er ikke så »kringlet«, som vi gerne vil have det til at være:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Så lad os prøve noget andet: at nedsætte antallet af døre. Vi forandrer 5-tallet i RND-sætningen i linie 20 til et 3-tal og får fx følgende resultat:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Nu er der alt, alt for mange »øer«, men til gengæld begynder kortet i strukturen at ligne den ikke alt for nemme labyrint, vi er ude efter. Er der virkelig ikke nogen mulighed for at fjerne disse »øer«?

Lad os forsøge:

```

210 FOR I=1 TO AL
220 IF I>89 AND L(I,1)=0 AND L(I,2)=0 AND L(I,3)=0
    AND L(I,4)=0 THEN 240
230 GOTO 250
240 L(I,1)=I-10:L(I-10,2)=I:GOTO 280

```

```

250 IF L(I,1)=0 AND L(I,2)=0 AND L(I,3)=0 AND
    L(I,4)=0 THEN 270
260 GOTO 280
270 L(I,2)=I+10:L(I+10,1)=I
280 NEXT I

```

Vi har nu sørget for, at der ikke kan opstå nogen enkeltrums-
»øer« på vores kort. Hvis vi igen bruger et treltal i RND-sætning-
en i linie 20 vil vores kort fx komme til at se sådan ud:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Stadig ikke godt. Strukturen er fin, og »øerne« er større, men de er der stadig. Vi må skille os af med dem på en eller anden

måde. Vi må prøve at *forbinde* dem med »hovedlandmasserne«. Hertil kræves et noget mere indviklet program, som du får her:

```
10 LO=1:GA=1
15 DIM V$(4),L(100,4),G(100,5)
20 FOR I=1 TO 100:FOR J=1 TO 4:L(I,J)=INT(RND(1)*2):
    NEXT J:NEXT I
30 FOR I=1 TO 10:L(I,1)=0:NEXT I
40 FOR I=91 TO 100:L(I,2)=0:NEXT I
50 FOR I=10 TO 100 STEP 10:L(I,3)=0:NEXT I
60 FOR I=1 TO 91 STEP 10:L(I,4)=0:NEXT I
70 FOR I=1 TO 100
80 IF I<11 THEN 100
90 IF L(I,1)=0 THEN L(I-10,2)=0
100 IF L(I,1)<>0 THEN L(I,1)=I-10
110 IF I>90 THEN 130
120 IF L(I,2)=0 THEN L(I+10,1)=0
130 IF L(I,2)<>0 THEN L(I,2)=I+10
140 IF I=100 THEN 160
150 IF L(I,3)=0 THEN L(I+1,4)=0
160 IF L(I,3)<>0 THEN L(I,3)=I+1
170 IF I=1 THEN 190
180 IF L(I,4)=0 THEN L(I-1,3)=0
190 IF L(I,4)<>0 THEN L(I,4)=I-1
200 NEXT I
210 FOR I=1 TO 100
220 IF I>89 AND L(I,1)=0 AND L(I,2)=0 AND L(I,3)=0
    AND L(I,4)=0 THEN 260
250 GOTO 270
260 L(I,1)=I-10:L(I-10,2)=I:GOTO 300
270 IF L(I,1)=0 AND L(I,2)=0 AND L(I,3)=0 AND
    L(I,4)=0 THEN 290
280 GOTO 300
290 L(I,2)=I+10:L(I+10,1)=I
```



```

300 NEXT I
310 FOR I=1 TO 100: FOR J=1 TO 4:G(I,J)=L(I,J):NEXT
    J:NEXT I
320 IF L0=1 THEN 380
330 FOR I=1 TO 91 STEP 10:IF I=GA THEN 360
340 NEXT I
350 GOTO 370
360 L(GA,1)=GA-10:L(GA-10,2)=GA:GOTO 380
370 L(GA,4)=GA-1:L(GA-1,3)=GA
380 FOR I=GA TO 100
390 G(I,5)=G(I,5)+1:C=0
400 IF G(I,5)>20 THEN 680
410 IF G(I,1)+G(I,2)+G(I,3)+G(I,4)=0 THEN 680
420 IF G(I,2)<>0 THEN 440
430 GOTO 480
440 FOR J=1 TO 4:IF G(I,J)=0 THEN C=C+1
450 NEXT J
460 IF C=3 THEN G(I+10,1)=0
470 I=I+10:GOTO 660
480 C=0:IF G(I,3)<>0 THEN 500
490 GOTO 540
500 FOR J=1 TO 4:IF G(I,J)=0 THEN C=C+1
510 NEXT J
520 IF C=3 THEN G(I+1,4)=0
530 I=I+1:GOTO 660
540 C=0:IF G(I,1)<>0 THEN 560
550 GOTO 600
560 FOR J=1 TO 4:IF G(I,J)=0 THEN C=C+1
570 NEXT J
580 IF C=3 THEN G(I-10,2)=0
590 I=I-10:GOTO 660
600 C=0:IF G(I,4)<>0 THEN 620
610 GOTO 660
620 FOR J=1 TO 4:IF G(I,J)=0 THEN C=C+1
630 NEXT J

```

```

640 IF C=3 THEN G(I-1,3)=0
650 I=I-1:REM IKKE NOGET TILFÆLDE, AT DER STAR 2
    I=I-1 HER OG I NÆSTE LINIE. DEN ØVERSTE HØRER
    TIL I DEN RUTINE, DER BEGYNDER I LINIE 600, MENS
    DEN ANDEN ER GENEREL *****
660 I=I-1
670 NEXT I
680 GA=GA+1
690 IF GA=101 THEN 720
700 IF G(GA,5)<>0 THEN 680
710 GOTO 330
720 END

```

Den første del af programmet er næsten identisk med det, jeg allerede har gennemgået. Bemærk dog, at vi i linie 10 definerer en ny variabel GA, og at vi i linie 15 DIMensionerer et nyt sæt G(), og at vi i linie 20 sætter tallet i RND-sætningen til 2. Det giver flere »øer«, ved vi, men bare vent, til du ser det endelige resultat!

Fra linie 310 og fremefter er programmet helt nyt. Det er i denne del af det, at vi scanner det »øhav«, vi har fået konstrueret via den første del af programmet. Dette »øhav« (som er frembragt af programmet til og med linie 300) kan fx se sådan ud:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Lad os nu se på, hvordan vi får det hele til at hænge sammen:

310: Her gør vi sættet $G()$ lig med sættet $L()$. Det er $L()$, vi skal bruge, når vi engang er færdige, men indtil da har vi brug for nogle arbejdsvariabler, og dem kalder vi altså $G()$. $G()$ har i øvrigt et ekstra element i sin anden dimension (se linie 15), og det er dette femte element, vi vil bruge til at fortælle maskinen, om den har været på stedet før.

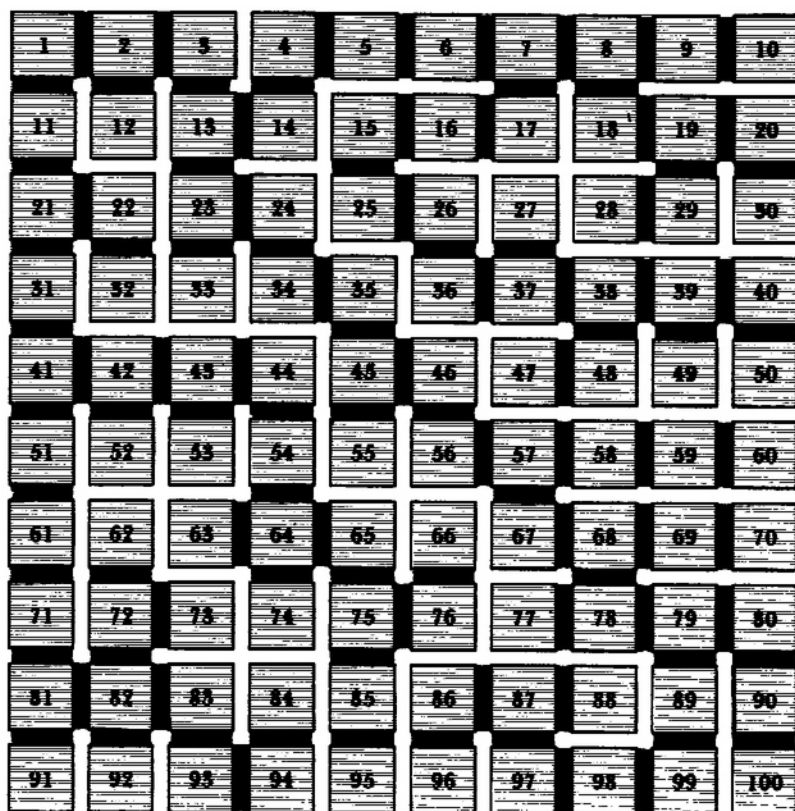
Vi har også brug for en arbejdsvariabel, der svarer til LO . Den har vi allerede defineret (linie 10) og givet navnet GA , så vi husker, at den hører til arbejds sættet $G()$.

320-370: Gemmer vi til senere. Se nedenfor.

380-720: Vi begynder forfra på den første »Ø« (se figur 10). Rum nr. 1 har kun én udgang, og efter at have givet $G(1,5)$ værdien $G(1,5)+1$ (dvs. 1, fordi $G(1,5)$ var lig nul, da vi startede), lukker vi af for rummet. Vi smider det simpelt hen ud af spillet, *men kun i vores arbejdsæt $G()$* . I sættet $G()$ vil det ikke længere eksistere som et rum med døre, og maskinen vil gå ned til Rum nr. 11, der er det eneste, som havde sammenhæng med Rum nr. 1. Også dette vil blive lukket ud af arbejdsættet $G()$, da det nu også kun har én udgang, fordi Rum nr. 1 er blevet registreret som ikke eksisterende. Derefter fortsættes til det næste rum, nr. 21, som ligeledes vil blive kasseret, fordi det nu kun har én udgang. »Øen«, der består af rummene 1, 11 og 21 eksisterer nu kun i sættet $G()$ som en tom enhed, der er blevet scannet og smidt væk – men kun næsten. For selv om de første fire elementer i den anden dimension i rummene 1, 11 og 21 nu er lig med nul, er den femte samtlige steder lig med mindst 1, hvilket fortæller maskinen, at den *har* været der. Men den kan ikke komme videre på denne »Ø«, eftersom den selv har lukket af for rummene et efter et, og derfor hopper den videre til den næste, dvs. til Rum nr. 2.

Her kommer linierne 320-370 ind i billedet. Maskinen ved nu, at rummene 1, 11 og 21 var en »Ø«, og nu giver vi den ordre til at forbinde denne »Ø« med den næste (den, den er på nu, rummene 2 og 12), men vi gør det *kun* i sættet $L()$, ikke i $G()$. I sættet $L()$ vil der være skabt den forbindelse mellem de to øer, som vi ønsker.

Maskinen kører med arbejdsættet $G()$ videre fra »Ø« til »Ø«, indtil den når Rum nr. 100, men den interesserer sig kun for de steder, hvor den endnu ikke har været (de steder, hvor $G(GA,5)=0$). Hver gang den kommer til en ny »Ø«, får den besked på i sættet $L()$ at forbinde denne med den forrige mod vest, undtagen i rummene 1 TO 91 STEP 10, hvor der jo ikke kan gås mod vest. Her går den mod nord, hvor den også ved, den har været. Når maskinen er kommet til Rum nr. 100, består kortet stadig væk af »øer« i sættet $G()$, men i sættet $L()$ vil disse »øer« være blevet kædet sammen, og det ovenstående kort vil have forvandlet sig til dette:



Her har vi vores sammenhængende landskab, der til forveksling ligner en labyrinth – 100 tilfældigt sammenkædede lokaliteter, der hænger sammen som en helhed, og som det ikke er helt nemt at hitte rundt i, hvis man ikke har et kort – 100 lokaliteter, som vi kan anbringe vores genstande og forhindringer i. Vi kan bruge disse 100 lokaliteter i et selvstændigt spil, eller vi kan bygge dem ind som en labyrinth i et andet spil, hvor resten af lokaliteterne er blevet kortlagt af os selv. Og tallet 100 er ikke nogen nødvendighed. Vi kan også lave en mindre labyrinth på 9 (3 gange 3) lokaliteter eller en stor en på 10.000 (100 gange 100 eller 10 gange 1000) lokaliteter – hvis vi vel at mærke har plads i maskinens hukommelse til det!

En labyrinth med 10.000 lokaliteter kunne spilleren bruge en rum (vitsen absolut utilsigtet!) tid på at udforske (og maskinen en

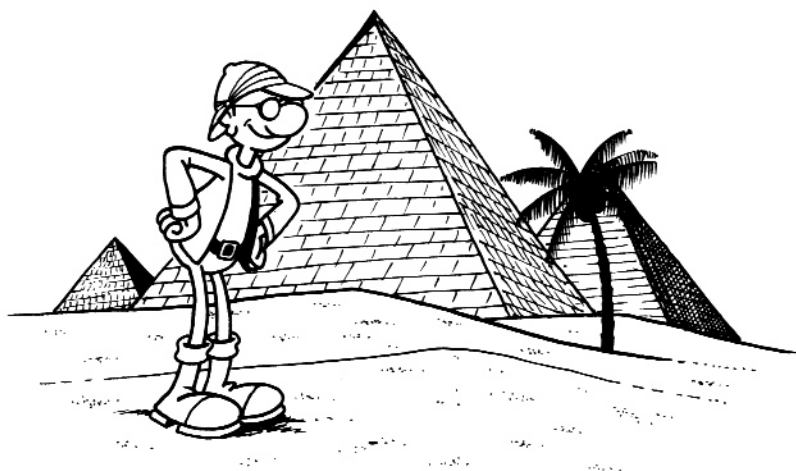
rum tid på at kortlægge). Spørgsmålet er, om det ville være særlig interessant, hvis der ikke kom variationer ind undervejs i form af lokalitetsbeskrivelser og genstande, der skulle bruges til et eller andet.

Men vent nu lige lidt! Lokalitetsbeskrivelser? Hvordan kan vi hæfte sådan nogen på vores labyrint (lad os holde os til den med de 100 rum), når vi ikke aner, hvordan rummene kommer til at se ud? Ville det ikke være ret fjollet at skrive »Du står i en blindgyde«, hvis der vitterlig er mulighed for at gå mod nord, syd og øst?

Er det et problem, der kan løses? Kan vi virkelig anbringe lokalitetsbeskrivelser mere eller mindre tilfældigt og alligevel få dem til at virke logiske?

Ja, det kan vi – mere herom i næste kapitel.

7 · Faraos skat



Lad os nu se på, hvordan vi kan bruge vores »lokalitets-generator« i praksis. En af de store fordele ved det »matematisk konstruerede« kort, som vi så i forrige kapitel, er, at det giver os mulighed for ligefrem at lade maskinen tegne et kort. Jeg sagde i forordet, at jeg ikk  ville bruge grafik-kommandoer (for at undg  den forvirring, konvertering til andre datamater end IBM PC eller Commodore ville medf re), og det *kan* faktisk (som du allerede s  i afsnittet »Korttegning« i kapitel 4) lade sig g re at tegne med computeren uden at bruge avancerede grafik-kommandoer. Men en kommando, som ikke findes i de helt simple BASIC-variationer, kan vi desv rre ikke komme helt udenom i det efterf lgende. Den hedder LOCATE p  Microsoft BASIC og PRINT AT p  fx Sinclair BASIC (og har du en Commodore-computer, skal du bruge en POKE-instruktion). Det er den instruktion, der g r, at man kan PRINTe et tegn et n rmere angivet sted p  sk rmen *oven i* noget, der er der i forvejen. De fleste BASIC-dialekter kan klare dette problem, men hvis *din* ikke kan,

skal du ikke fortvivle. Du kan sagtens spille »Faraos skat« uden – selv om det bliver i en lidt skrabet version. Men herom senere – dette blot for at fortælle dig, at du ikke skal sukke opgivende og lægge bogen væk, dersom du ikke har denne mulighed.

Lad os allerførst kigge på, hvad spillet går ud på – og derefter på, hvordan vi løser de problemer, der dukker op i forbindelse med konstrueringen.

Vi har allerede (i forrige kapitel) set, hvordan vi med ganske enkle BASIC-instruktioner kan bygge en labyrint på 100 lokaliteter, der alle hænger sammen. Lad os nu prøve at få et spil ud af det.

Lad os starte i Rum nr. 1 og vedtage, at vi skal finde en skat, der befinder sig i den anden ende af »landskabet«. Vi kunne anbringe den i Rum nr. 100, men så ved vi præcis, hvor den er og behøver sådan set ikke lede nærmere efter den. Vi kunne også anbringe den tilfældigt et hvilket som helst sted, men så risikerede vi, at den kom til at ligge i fx Rum nr. 2, og så var den skattejagt overstået efter første træk. Så lad os lægge den et sted i labyrintens sydlige ende – et sted i »90'erne«. Det gør den en anelse sværere at finde, og så har vi en garanti for, at den ikke ligger lige ved siden af udgangspunktet.

Men eftersom vi ved, at vores kort hænger sammen, ved vi også, at spilleren med usvigelig sikkerhed på et eller andet tidspunkt vil finde skatkammeret, hvis han leder ihærdigt nok – og det kan i nogle tilfælde faktisk gå ret hurtigt. Så lad os opstille nogle betingelser, der skal opfyldes, før han kan tage skatten – også selv om han ved, hvor den ligger. Vi kunne som sædvanlig lave et genstandssæt som i Amazon-spillet, indbygge rifler og ammunition, leoparder og nøgler (det synes jeg personligt, du selv skal gøre, når du laver dit eget spil), men lad os for at gøre demonstrationen så enkel som mulig prøve noget andet:

Vi vedtager, at skatten ligger skjult bag et panel, der kun kan glide til side, hvis det påvirkes af den samlede kraft fra ti magnetsten, der ligger tilfældigt anbragt rundt omkring i labyrinten – der i dette spil er blevet til en pyramide, som skal udforskes. For at få fat i skatten skal spilleren indsamle alle ti magnetsten, og det vil i praksis betyde, at han bliver nødt til at udforske *hele* labyrinten, før han kan nå til vejs ende.

Lad os gøre det endnu sværere for ham. Lad os tage magnetstenene fra ham, når han mindst venter det, så han er nødt til at begynde forfra – naturligvis inden for rimelighedens grænser, så han ikke bliver træt og giver op. Lad os til dette formål prøve at bruge den anagram-rutine, vi kiggede på i kapitel 4.

Vi anbringer i pyramiden en mumie, der nu og da dukker op og forlanger at få løsningen på en forholdsvis simpel anagram-opgave. Svares der forkert, tager han som straf magnetstenene og lægger dem på nye, tilfældige steder – og så må den stakkels spiller begynde forfra med indsamlingen. Lad os lade mumien forlange, at spilleren siger hans navn, som er et anagram af tre opgivne bogstaver. Det vil give spilleren én ud af fem muligheder for at svare rigtigt (for den kombination, mumien opgiver, er naturligvis ikke den rigtige) – og hvis han ellers holder styr på sine svar, vil han kun kunne blive generet af mumien højest fem gange, før han svarer rigtigt.

Lad os også udnytte muligheden for at få tegnet et kort og et sted i pyramiden anbringe et magisk spejl, hvori man kan se et sådant kort.

Opgaven går altså ud på at finde skatten, finde ti magnetsten og finde ud af mumiens navn. Ret overskueligt, men glem ikke, at dette er et demonstrationsprogram, og at der ikke er nogen, der forhindrer dig i at gøre livet endnu mere fortrædeligt for den stakkels spiller, når du laver dine egne programmer.

Lad os på dette punkt kigge på selve programmet:

```

19 REM FARAOS SKAT *****
20 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET:PRINT
   CHR$(147) *****
30 GOSUB 3620:REM NED TIL INDLEDNING *****
49 REM DATA *****
50 LO=1:GA=1:MU=-1:S$=CHR$(13)
60 DIM V$(4),L(100,4),G(100,5),S(10),S$(10),LO$(40),
   L$(100)
70 FOR I=1 TO 4:READ V$(I):NEXT I
80 FOR I=1 TO 10:READ S$(I):NEXT I

```

```

90 DATA NORDPA,SYDPA,ØSTPA,VESTPA
100 DATA FØRSTE,ANDEN,TREDJE,FJERDE,FEMTE,SJETTE,
    SYVENDE,OTTENDE,NIENDE,TIENDE
110 REM BEREGNING AF GRUNDTAL TIL
    TILFÆLDIGHEDSTALSGENERATOREN *****
120 T=(VAL(RIGHT$(TIME$,2))+1)*
    (VAL(MID$(TIME$,4,2)))*(VAL(LEFT$(DATE$,2))):
    REM COMMODORE-BRUGER, SPRING DENNE LINIE OVER
    *****
130 RANDOMIZE T:REM COMMODORE-BRUGER, SPRING DENNE
    LINIE OVER *****
149 REM BEREGNING AF SKATTENS OG MAGNETSTENENES
    POSITION *****
150 SK=INT(RND(1)*9)+92
160 FOR I=1 TO 10:S(I)=INT(RND(1)*99)+2
170 FOR J=0 TO I-1
180 IF S(I)=S(J) OR S(I)=SK THEN
    S(I)=INT(RND(1)*99)+2:GOTO 170
190 NEXT J:NEXT I
209 REM FØRSTE KORTLÆGNING AF LABYRINTEN *****
210 FOR I=1 TO 100:FOR J=1 TO 4:
    L(I,J)=INT(RND(1)*2):NEXT J:NEXT I
220 FOR I=1 TO 10:L(I,1)=0:NEXT I
230 FOR I=91 TO 100:L(I,2)=0:NEXT I
240 FOR I=10 TO 100 STEP 10:L(I,3)=0:NEXT I
250 FOR I=1 TO 91 STEP 10:L(I,4)=0:NEXT I
260 FOR I=1 TO 100
270 IF I<11 THEN 290
280 IF L(I,1)=0 THEN L(I-10,2)=0
290 IF L(I,1)<>0 THEN L(I,1)=I-10
300 IF I>90 THEN 320
310 IF L(I,2)=0 THEN L(I+10,1)=0
320 IF L(I,2)<>0 THEN L(I,2)=I+10
330 IF I=100 THEN 350
340 IF L(I,3)=0 THEN L(I+1,4)=0

```

```

350 IF L(I,3)<>0 THEN L(I,3)=I+1
360 IF I=1 THEN 380
370 IF L(I,4)=0 THEN L(I-1,3)=0
380 IF L(I,4)<>0 THEN L(I,4)=I-1
390 NEXT I
400 FOR I=1 TO 100
410 IF I>89 AND L(I,1)=0 AND L(I,2)=0 AND L(I,3)=0
    AND L(I,4)=0 THEN 430
420 GOTO 440
430 L(I,1)=I-10:L(I-10,2)=I:GOTO 470
440 IF L(I,1)=0 AND L(I,2)=0 AND L(I,3)=0 AND
    L(I,4)=0 THEN 460
450 GOTO 470
460 L(I,2)=I+10:L(I+10,1)=I
470 NEXT I
489 REM SCANNING AF LABYRINTEN. "ØERNE" FORBINDES
    MED HINANDEN *****
490 PRINT:PRINT "VÆR TALMODIG, VÆR TALMODIG!"
500 FOR I=1 TO 100: FOR J=1 TO 4:G(I,J)=L(I,J):
    NEXT J:NEXT I
510 IF LO=1 THEN 570
520 FOR I=1 TO 91 STEP 10:IF I=GA THEN 550
530 NEXT I
540 GOTO 560
550 L(GA,1)=GA-10:L(GA-10,2)=GA:GOTO 570
560 L(GA,4)=GA-1:L(GA-1,3)=GA
570 FOR I=GA TO 100
580 G(I,5)=G(I,5)+1:C=0
590 IF G(I,5)>20 THEN 870
600 IF G(I,1)+G(I,2)+G(I,3)+G(I,4)=0 THEN 870
610 IF G(I,2)<>0 THEN 630
620 GOTO 670
630 FOR J=1 TO 4:IF G(I,J)=0 THEN C=C+1
640 NEXT J
650 IF C=3 THEN G(I+10,1)=0

```

```

660 I=I+10:GOTO 850
670 C=0:IF G(I,3)<>0 THEN 690
680 GOTO 730
690 FOR J=1 TO 4:IF G(I,J)=0 THEN C=C+1
700 NEXT J
710 IF C=3 THEN G(I+1,4)=0
720 I=I+1:GOTO 850
730 C=0:IF G(I,1)<>0 THEN 750
740 GOTO 790
750 FOR J=1 TO 4:IF G(I,J)=0 THEN C=C+1
760 NEXT J
770 IF C=3 THEN G(I-10,2)=0
780 I=I-10:GOTO 850
790 C=0:IF G(I,4)<>0 THEN 810
800 GOTO 850
810 FOR J=1 TO 4:IF G(I,J)=0 THEN C=C+1
820 NEXT J
830 IF C=3 THEN G(I-1,3)=0
840 I=I-1:REM IKKE NOGET TILFÆLDE, AT DER STAR 2
      I=I-1 HER OG I NÆSTE LINIE. DEN ØVERSTE HØRER
      TIL I DEN RUTINE, DER BEGYNDER I LINIE 790, MENS
      DEN ANDEN ER GENEREL *****
850 I=I-1
860 NEXT I
870 GA=GA+1
880 IF GA=101 THEN 910
890 IF G(GA,5)<>0 THEN 870
900 GOTO 520
910 PRINT:PRINT "NU VARER DET IKKE SA LÆNGE."
929 REM LOKALITETSBeregning *****
930 LO$(1)="LIGE INDEN FOR"+S$+"INDGANGEN TIL
      PYRAMIDEN."
940 LO$(2)="I ET STORT, TOMT RUM."
950 LO$(3)="I EN LIGE GANG."
960 LO$(4)="I EN GANG"+S$+"DER SLAR ET KNÆK."

```

970 LO\$(5)="I EN BLINDGYDE."
 980 LO\$(6)="I SKATKAMMERET."
 990 L1\$="ET KORT OVER PYRAMIDEN."
 1000 LO\$(7)="FORAN ET MAGISK SPEJL,"+S\$+"I HVILKET
 DU KAN SE"+S\$+L1\$
 1010 LO\$(8)="I ET RUM"+S\$+"FYLDT MED STØVEDE
 STATUER."
 1020 LO\$(9)="I ET RUM"+S\$+"FYLDT MED MØRNEDE
 PAPYRUSRULLER."
 1030 LO\$(10)="I ET RUM"+S\$+"FYLDT MED POTTESKAR."
 1040 LO\$(11)="I ET STORT RUM"+S\$+"DER MYLDRER MED
 ROTTER."
 1050 LO\$(12)="FORAN EN TOM MUMIEKISTE."
 1060 LO\$(13)="I ET RUM"+S\$+"MED SMUKKE VÆGMALERIER."
 1070 LO\$(14)="I ET RUM"+S\$+"MED INDHUGGEDE
 HIEROGLYFFER."
 1080 LO\$(15)="I ET RUM"+S\$+"FYLDT MED RUSTNE VÅBEN."
 1090 LO\$(16)="FORAN"+S\$+"FARAO TUT LI HUTS
 SARKOFAG."
 1100 LO\$(17)="I ET RUM"+S\$+"MED SKRA VÆGGE."
 1110 LO\$(18)="I ET RUM"+S\$+"FYLDT MED SKELETTER."
 1120 LO\$(19)="I ET RUM"+S\$+"MED GAMLE VASER OG
 KRUKKER."
 1130 LO\$(20)="I ET RUM"+S\$+"HVOR DER HÆNGER"+S\$+"EN
 UDBRÆNDT FAKKEL PÅ VÆGGEN."
 1140 L2\$="FOR AT KOMME IGENNEM DET."
 1150 LO\$(21)="I ET KAMMER,"+S\$+"DER ER SÅ LAVT, AT
 DU MA BØJE DIG"+S\$+L2\$
 1160 L3\$="AF GUDERNE KHNOL OG THOT."
 1170 LO\$(22)="I ET AFLANGT KAMMER"+S\$+"MED TO
 VELBEVAREDE STATUER"+S\$+L3\$
 1180 LO\$(23)="FORAN EN STENREOL MED"+S\$+"EN
 INTERESSANT SAMLING AF DØDEBØGER."
 1190 LO\$(24)="FORAN EN HALVRADDEN LÆDERSÆK"+S\$+"MED
 GAMLE KOBBERMØNTER."

1200 L0\$(25)="I ET RUM"+S\$+"FYLDT MED INDTØRREDE
 KATTEMUMIER."
 1210 L9\$="RHIP, RHAP OG RHUP."
 1220 L0\$(26)="FORAN MUMIERNE"+S\$+"AF DE TRE SMA
 PRINSER+S\$+L9\$
 1230 L4\$="FARAO OSMOSES IIIs INDSKRUMPNE BLINDTARM."
 1240 L0\$(27)="FORAN EN URNE MED"+S\$+L4\$
 1250 L5\$="EN OLDÆGYPTISK FOLKESANG AF
 DEN"+S\$+"BERØMTE SKJALD IENHS MEMPHIS."
 1260 L0\$(28)="FORAN EN"+S\$+"HIEROGLYF-FRISE
 MED"+S\$+L5\$
 1270 L6\$="OLDÆGYPTISKE LÆSKEDRIK KOGHAKOHLAH."
 1280 L0\$(29)="FORAN ET STENFAD"+S\$+"MED INDTØRREDE
 RESTER AF DEN"+S\$+L6\$
 1290 L7\$="KAHFEGRUMSES IV."
 1300 L0\$(30)="FORAN ET RELIEF"+S\$+"AF FARAO TUT LI
 HUTS FAR,"+S\$+L7\$
 1310 L0\$(31)="FORAN EN FORBINDSKASSE"+S\$+"MED
 UBRUGTE MUMIEBANDAGER."
 1320 L0\$(32)="I ET RUM"+S\$+"MED ET NYDELIGT
 FLISEGULV."
 1330 L0\$(33)="I DET, DER ENGANG VAR"+S\$+
 "PYRAMIDEARBEJDERNES KANTINE."
 1340 L0\$(34)="I EN TRANG CELLE"+S\$+"MED LÆNKER
 FASTGJORT TIL VÆGGEN."
 1350 L0\$(35)="I ET STORT RUM"+S\$+"MED ET FORSÆNKET
 OFFERKAR I GULVET."
 1360 L0\$(36)="FORAN EN SAMLING"+S\$+"AF UDSTOPPEDE
 SLAVER."
 1370 L8\$="HARSK OG ILDELUGTENDE SALVE."
 1380 L0\$(37)="VED EN HYLDE"+S\$+"MED NOGLE KRUKKER
 FYLDT MED"+S\$+L8\$
 1390 L0\$(38)="FORAN EN VÆG"+S\$+"HVORPA DER STAR
 'AKHNATON WAS HERE'. "
 1400 L0\$(39)=S\$+"I ET STORT HØJLOFTET RUM."

```

1410 L0$(40)="FORAN EN"+S$+"TILSYNELADENDE BUNDLØS
      BRØND."
1420 L$(1)=L0$(1)
1430 FOR I=2 TO 100
1440 IF L(I,1)<>0 AND L(I,2)<>0 THEN L$(I)=L0$(3)
1450 IF L(I,3)<>0 AND L(I,4)<>0 THEN L$(I)=L0$(3)
1460 IF (L(I,1)<>0 AND L(I,3)<>0) OR (L(I,1)<>0 AND
      L(I,4)<>0) THEN L$(I)=L0$(4)
1470 IF (L(I,2)<>0 AND L(I,3)<>0) OR (L(I,2)<>0 AND
      L(I,4)<>0) THEN L$(I)=L0$(4)
1480 AA=0
1490 FOR J=1 TO 4
1500 IF L(I,J)=0 THEN AA=AA+1
1510 NEXT J
1520 IF AA=3 THEN L$(I)=L0$(5)
1530 IF AA=0 OR AA=1 THEN L$(I)=L0$(2)
1540 L$(SK)=L0$(6)
1550 NEXT I
1560 PRINT:PRINT "NU ER DET LIGE FØR!"
1570 FOR I=1 TO 100
1580 IF L$(I)=L0$(2) THEN LG=LG+1
1590 NEXT I
1600 FOR J=7 TO 40
1610 IF J=LG+7 THEN J=40:GOTO 1680
1620 AB=INT(RND(1)*99)+2
1630 IF L$(AB)<>L0$(2) THEN J=J-1:GOTO 1680
1640 FOR K=6 TO J-1
1650 IF L$(AB)=L0$(K) THEN AB=INT(RND(1)*99)+2:GOTO
      1640
1660 L$(AB)=L0$(J)
1670 NEXT K
1680 NEXT J
1690 GOSUB 4360:REM NED TIL
      MELLEMRUMSTANGENTS-RUTINE *****
1700 GOSUB 3810:A=1:GOTO 2160:REM NED TIL RESTEN AF

```

```

      INDLEDNINGEN, SOM ER LIG HJÆLPE-RUTINEN *****
1719 REM BEREGNING AF MUMIENS LOKALITET *****
1720 IF S>1 AND NA<>2 THEN MU=INT(RND(1)*40)
1730 IF MU=0 THEN GOSUB 4580:REM NED TIL
      MUMIE-RUTINE *****
1740 GOSUB 2110:REM NED TIL SKÆRMTEKST-RUTINE *****
1750 REM HER CHECKES, OM DER ER EN MAGNETSTEN PÅ
      STEDET *****
1760 FOR I=1 TO 10:IF S(I)=LO THEN 1790
1770 NEXT I
1780 GOTO 1810
1790 GOSUB 4530:GOTO 1910:REM NED TIL
      MAGNETSTEN-RUTINE *****
1809 REM SKATKAMMER-RUTINE *****
1810 IF L$(LO)<>LO$(6) THEN 1910
1820 IF S=10 THEN 1860
1830 PRINT:PRINT "DU HAR IKKE ALLE TI MAGNETSTEN,"
1840 PRINT "SA DU KAN IKKE AKTIVERE"
1850 PRINT "DEN HEMMELIGE MEKANISME.":GOTO 1910
1860 PRINT:PRINT "MAGNETSTENENES SAMLEDE KRAFT"
1870 PRINT "AKTIVERER ET HEMMELIGT PANEL,":PRINT
      "DER LANGSOMT GLIDER TIL SIDE."
1880 PRINT:PRINT "DU HAR FUNDET SKATTEN!":PRINT "TIL
      LYKKE!"
1890 K=1:FOR I=1 TO 6000:NEXT I:GOTO 2250
1909 REM INPUT-RUTINE *****
1910 PRINT:PRINT "HVAD VIL DU GØRE? ";
1920 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I
      STEDET:GET A$ *****
1925 IF A$="" THEN 1920
1930 PRINT A$
1940 NO=0
1950 PRINT
1960 IF A$="N" THEN NO=1
1970 IF A$="S" THEN NO=2

```




```
1980 IF A$="Ø" THEN NO=3
1990 IF A$="V" THEN NO=4
2000 IF A$="K" THEN 2240
2010 IF A$="H" THEN GOSUB 3810:GOTO 1720
2020 IF A$="U" THEN K=1:GOTO 2250
2030 IF NO=0 THEN PRINT "OM IGEN!":GOTO 1910
2040 IF LO<>0 THEN 2070
2050 IF NO<>3 THEN PRINT "GA NU ØSTPA, FOR POKKER!":
      GOTO 1910
2060 LO=1:GOTO 1720
2070 IF LO=1 AND NO=4 THEN LO=0:GOTO 1720
```

```

2080 IF L(LO,NO)=0 THEN PRINT "DU KAN IKKE GA DEN
      VEJ.":GOTO 1910
2090 LO=L(LO,NO):NO=0:GOTO 1720
2109 REM SKÆRMTEKST-RUTINE *****
2110 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
2120 IF LO<>0 THEN 2160
2130 PRINT "DU ER TILBAGE, HVOR DU STARTEDE."
2140 PRINT:PRINT "HVIS DU VIL IND I PYRAMIDEN IGEN,"
2150 PRINT "SKAL DU GA ØSTPA.":GOTO 1910
2160 PRINT "DU STAR "+L$(LO):PRINT
2170 PRINT "DU KAN GA:"
2180 IF LO=1 THEN PRINT "VESTPA"
2190 FOR I=1 TO 4:IF L(LO,I)<>0 THEN PRINT V$(I)
2200 NEXT I
2210 IF A=1 THEN A=2:GOTO 1910
2220 RETURN
2239 REM KORTEGNINGS-RUTINE *****
2240 IF L$(LO)<>LO$(7) THEN PRINT "DER ER IKKE NOGET
      KORT HER.":GOTO 1910
2250 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
2260 X$=CHR$(219)+CHR$(219):REM COMMODORE-BRUGER,
      SKRIV I STEDET: X$=CHR$(166) *****
2270 W1$="      ":REM COMMODORE-BRUGER, SKRIV I STEDET:
      W1$=" " *****
2280 W2$="      ":REM COMMODORE-BRUGER, SKRIV I
      STEDET: W2$=" " *****
2290 Y$=X$+CHR$(219):REM COMMODORE-BRUGER, SKRIV I
      STEDET: Y$=X$ *****
2300 PRINT:REM COMMODORE-BRUGER, STRYG DENNE LINIE
      *****
2310 PRINT TAB(10):FOR I=1 TO 10
2320 IF L(I,3)=0 THEN 2340
2330 PRINT X$;Y$;:GOTO 2350

```

```

2340 PRINT X$;W1$;
2350 IF I=10 THEN PRINT
2360 NEXT I
2370 PRINT TAB(10):FOR I=11 TO 20
2380 IF I=L((I-10),2) THEN 2400
2390 PRINT W2$;:GOTO 2410
2400 PRINT X$;W1$;
2410 IF I=20 THEN PRINT
2420 NEXT I
2430 PRINT TAB(10):FOR I=11 TO 20
2440 IF L(I,3)=0 THEN 2460
2450 PRINT X$;Y$;:GOTO 2470
2460 PRINT X$;W1$;
2470 IF I=20 THEN PRINT
2480 NEXT I
2490 PRINT TAB(10):FOR I=21 TO 30
2500 IF I=L((I-10),2) THEN 2520
2510 PRINT W2$;:GOTO 2530
2520 PRINT X$;W1$;
2530 IF I=30 THEN PRINT
2540 NEXT I
2550 PRINT TAB(10):FOR I=21 TO 30
2560 IF L(I,3)=0 THEN 2580
2570 PRINT X$;Y$;:GOTO 2590
2580 PRINT X$;W1$;
2590 IF I=30 THEN PRINT
2600 NEXT I
2610 PRINT TAB(10):FOR I=31 TO 40
2620 IF I=L((I-10),2) THEN 2640
2630 PRINT W2$;:GOTO 2650
2640 PRINT X$;W1$;
2650 IF I=40 THEN PRINT
2660 NEXT I
2670 PRINT TAB(10):FOR I=31 TO 40
2680 IF L(I,3)=0 THEN 2700

```

```

2690 PRINT X$;Y$;:GOTO 2710
2700 PRINT X$;W1$;
2710 IF I=40 THEN PRINT
2720 NEXT I
2730 PRINT TAB(10):FOR I=41 TO 50
2740 IF I=L((I-10),2) THEN 2760
2750 PRINT W2$;:GOTO 2770
2760 PRINT X$;W1$;
2770 IF I=50 THEN PRINT
2780 NEXT I
2790 PRINT TAB(10):FOR I=41 TO 50
2800 IF L(I,3)=0 THEN 2820
2810 PRINT X$;Y$;:GOTO 2830
2820 PRINT X$;W1$;
2830 IF I=50 THEN PRINT
2840 NEXT I
2850 PRINT TAB(10):FOR I=51 TO 60
2860 IF I=L((I-10),2) THEN 2880
2870 PRINT W2$;:GOTO 2890
2880 PRINT X$;W1$;
2890 IF I=60 THEN PRINT
2900 NEXT I
2910 PRINT TAB(10):FOR I=51 TO 60
2920 IF L(I,3)=0 THEN 2940
2930 PRINT X$;Y$;:GOTO 2950
2940 PRINT X$;W1$;
2950 IF I=60 THEN PRINT
2960 NEXT I
2970 PRINT TAB(10):FOR I=61 TO 70
2980 IF I=L((I-10),2) THEN 3000
2990 PRINT W2$;:GOTO 3010
3000 PRINT X$;W1$;
3010 IF I=70 THEN PRINT
3020 NEXT I
3030 PRINT TAB(10):FOR I=61 TO 70

```

```

3040 IF L(I,3)=0 THEN 3060
3050 PRINT X$;Y$;:GOTO 3070
3060 PRINT X$;W1$;
3070 IF I=70 THEN PRINT
3080 NEXT I
3090 PRINT TAB(10):FOR I=71 TO 80
3100 IF I=L((I-10),2) THEN 3120
3110 PRINT W2$;:GOTO 3130
3120 PRINT X$;W1$;
3130 IF I=80 THEN PRINT
3140 NEXT I
3150 PRINT TAB(10):FOR I=71 TO 80
3160 IF L(I,3)=0 THEN 3180
3170 PRINT X$;Y$;:GOTO 3190
3180 PRINT X$;W1$;
3190 IF I=80 THEN PRINT
3200 NEXT I
3210 PRINT TAB(10):FOR I=81 TO 90
3220 IF I=L((I-10),2) THEN 3240
3230 PRINT W2$;:GOTO 3250
3240 PRINT X$;W1$;
3250 IF I=90 THEN PRINT
3260 NEXT I
3270 PRINT TAB(10):FOR I=81 TO 90
3280 IF L(I,3)=0 THEN 3300
3290 PRINT X$;Y$;:GOTO 3310
3300 PRINT X$;W1$;
3310 IF I=90 THEN PRINT
3320 NEXT I
3330 PRINT TAB(10):FOR I=91 TO 100
3340 IF I=L((I-10),2) THEN 3360
3350 PRINT W2$;:GOTO 3370
3360 PRINT X$;W1$;
3370 IF I=100 THEN PRINT
3380 NEXT I

```

```

3390 PRINT TAB(10):FOR I=91 TO 100
3400 IF L(I,3)=0 THEN 3420
3410 PRINT X$;Y$;:GOTO 3430
3420 PRINT X$;W1$;
3430 IF I=100 THEN PRINT
3440 NEXT I
3450 X$=CHR$(176)+CHR$(176):REM COMMODORE-BRUGER,
      SKRIV I STEDET: X$=CHR$(113) *****
3460 Y$=CHR$(177)+CHR$(177):REM COMMODORE-BRUGER,
      SKRIV I STEDET: Y$=CHR$(122) *****
3470 Z$=CHR$(178)+CHR$(178):REM COMMODORE-BRUGER,
      SKRIV I STEDET: Z$=CHR$(119) *****
3480 GOSUB 4400
3485 LOCATE P,Q:PRINT X$:REM COMMODORE-BRUGER, SKRIV
      I STEDET: POKE P*40+Q+983,81 *****
3490 MI=LO
3500 FOR I=1 TO 10:LO=S(I):GOSUB 4400
3510 IF LO<>0 THEN LOCATE P,Q:PRINT Y$:REM
      COMMODORE-BRUGER, SKRIV I STEDET: IF LO<>0 THEN
      POKE P*40+Q+983,87 *****
3520 NEXT I
3530 LO=SK:GOSUB 4400
3535 LOCATE P,Q:PRINT Z$:REM COMMODORE-BRUGER, SKRIV
      I STEDET: POKE P*40+Q+983,90 *****
3540 IF K=1 THEN PRINT "SLUT,":GOTO 5230
3550 LO=MI
3560 PRINT:PRINT TAB(10) X$;" : DIG.    "{Z$;" : SKAT-
      TEN.";;REM COMMODORE-BRUGER, SKRIV I STEDET:
      PRINT:PRINT X$;" : DIG.    "{Y$;" : SKATTEN.  "{
      *****
3570 PRINT "{Y$;" : MAGNETSTENENE.";REM
      COMMODORE-BRUGER: SKRIV I STEDET: PRINT Z$;
      ": MAGNETSTENENE."
3580 PRINT:PRINT TAB(10)"TRYK PA 'F', NAR DU VIL
      FORTSÆTTE."

```

```

3590 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
      GET A$ *****
3595 IF A$<>"F" THEN 3590
3600 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
3605 GOTO 1720
3619 REM INDLEDNING *****
3620 PRINT "*****"
3630 PRINT:PRINT "          FARAOS SKAT"
3640 PRINT:PRINT "ET EVENTYRSPIL FOR ÆGYPTOLOGER"
3650 PRINT
3660 PRINT "*****"
3670 PRINT
3680 PRINT "DU STAR UDEN FOR INDGANGEN TIL"
3690 PRINT "FARAO TUT LI HUTS PYRAMIDE."
3700 PRINT "INDGANGEN BEFINDER SIG I PYRAMIDENS"
3710 PRINT "NORDVESTLIGSTE HJØRNE, OG DU VED,"
3720 PRINT "AT DER ER ET SKATKAMMER I"
3730 PRINT "PYRAMIDENS SYDLIGE ENDE."
3740 PRINT "FOR AT NA DETTE SKATKAMMER"
3750 PRINT "SKAL DU GENNEM PYRAMIDENS":PRINT
      "KRINGLEDE GANGE."
3760 PRINT:PRINT "VENT ET ØJEBLIK, MENS MASKINEN"
3770 PRINT "KORTLÆGGER PYRAMIDEN, DER"
3780 PRINT "ALDRIG BLIVER DEN SAMME TO GANGE."
3790 RETURN
3809 REM HJÆLPE-RUTINE *****
3810 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
3820 PRINT "ET STED INDE I PYRAMIDEN"
3830 PRINT "ER DER EFTER SIGENDE ET MAGISK SPEJL,"
3840 PRINT "SOM VISER ET KORT OVER PYRAMIDEN."
3850 PRINT "MEN DER ER INGEN, DER SIGER,"
3860 PRINT "AT DET ER NEMT AT FINDE,"
3870 PRINT "OG DU HAR OGSÅ EN UBEHAGELIG"

```

```

3880 PRINT "FORNEMMELSE AF, AT DER LURER"
3890 PRINT "UKENDTE FARER I MØRKET"
3900 PRINT "INDE I DEN SINDRIGT KONSTRUERED"
3910 PRINT "LABYRINT, SOM FARAO TUT LI HUTS"
3920 PRINT "NAVNKUNDIGE BYGMESTER ARK ITEKHT"
3930 PRINT "I SIN TID OPFØRTE I PYRAMIDENS INDRE"
3940 PRINT "FOR AT FORHINDRE NEDRIGE GRAVRØVERE"
3950 PRINT "SOM DIG I AT FA FAT I SKATTEN."
3960 GOSUB 4360
3970 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
3980 PRINT "DU SKAL I DETTE SPIL BRUGE"
3990 PRINT "ENKELTBOGSTAVS-KOMMANDOER."
4000 PRINT "DE ER SOM FØLGER:"
4010 PRINT "'H' FOR 'HJÆLP'"
4020 PRINT "(KALDER DENNE VEJLEDNING FREM IGEN),"
4030 PRINT "'N' FOR 'GA NORDPA',"
4040 PRINT "'S' FOR 'GA SYDPA',"
4050 PRINT "'Ø' FOR 'GA ØSTPA'"
4060 PRINT "'V' FOR 'GA VESTPA',"
4070 PRINT "'K' FOR 'KIGGE PÅ KORTET'"
4080 PRINT "SAMT 'U' FOR 'UD', HVIS DU"
4090 PRINT "ØNSKER AT STOPPE SPILLET."
4100 PRINT "KUN HVIS DU BLIVER BEDT OM ET HELT ORD,"
4110 PRINT "SKAL DU BAGEFTER TRYKKE PÅ
      RETUR-TASTEN."
4120 PRINT "ELLERS ER DET KUN TASTERNE MED DE"
4130 PRINT "OVENNÆVNTE BOGSTAVER, DU SKAL TRYKKE
      PÅ."
4140 GOSUB 4360
4150 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
4160 PRINT "DET ER IMIDLERTID IKKE NOK,"
4170 PRINT "AT DU FINDER SKATKAMMERET,"
4180 PRINT "FOR SKATTEN ER GEMT BAG ET"

```



```

4190 PRINT "HEMMELIGT PANEL, DER KUN KAN"
4200 PRINT "AKTIVERES AF DEN SAMLEDE KRAFT"
4210 PRINT "FRA TI MAGNETSTEN, SOM LIGGER"
4220 PRINT "TI FORSKELLIGE STEDER I PYRAMIDEN."
4230 PRINT "FOR AT FÅ FAT I SKATTEN ER DU"
4240 PRINT "ALTSÅ FØRST NØDT TIL AT"
4250 PRINT "FINDE DISSE TI MAGNETSTEN."
4260 PRINT "SKATTENS, MAGNETSTENENES OG"
4270 PRINT "DIN EGEN POSITION ER ANGIVET"
4280 PRINT "PÅ DET KORT, DU MASKE FINDER."
4290 PRINT "DU VIL KUNNE KIGGE PÅ DET"
4300 PRINT "LIGE SÅ LÆNGE, DU VIL, MEN DU"
4310 PRINT "KAN IKKE TAGÉ DET MED DIG!"
4320 GOSUB 4360
4330 IF A=2 THEN RETURN
4340 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
4345 PRINT "DU GÅR IND I PYRAMIDEN.":PRINT:RETURN
4350 REM MELLEMRUMSTANGENT-RUTINE *****
4360 PRINT:PRINT "TRYK PÅ MELLEMRUMSTASTEN,":PRINT
      "NÅR DU VIL FORTSÆTTE."
4370 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
      GET A$ *****
4375 IF A$(">") THEN 4370
4380 RETURN
4399 REM BEREKNING AF LOKALITET TIL PRINTNING PÅ
      KORTET *****
4400 IF LO<11 THEN P=2:GOTO 4500
4410 IF LO<21 THEN P=4:GOTO 4500
4420 IF LO<31 THEN P=6:GOTO 4500
4430 IF LO<41 THEN P=8:GOTO 4500
4440 IF LO<51 THEN P=10:GOTO 4500
4450 IF LO<61 THEN P=12:GOTO 4500
4460 IF LO<71 THEN P=14:GOTO 4500
4470 IF LO<81 THEN P=16:GOTO 4500

```

```

4480 IF LO<91 THEN P=18:GOTO 4500
4490 P=20
4500 Q=5*(LO-(((P/2)-1)*10))+5:REM COMMODORE-BRUGER,
      SKRIV I STEDET: Q=2*(LO-(((P/2)-1)*10))+9 *****
4510 RETURN
4529 REM MAGNETSTENS-RUTINE *****
4530 S(I)=0:S=S+1:PRINT:PRINT "PA DETTE STED FINDER
      DU"
4540 PRINT "DEN ";S$(S);" MAGNETSTEN."
4550 PRINT "DU SAMLER DEN OP OG TAGER DEN MED."
4560 RETURN
4579 REM MUMIE-RUTINE *****
4580 IF NA=0 THEN 4620
4590 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
4595 PRINT "PLUDELIG STAR MUMIEN FORAN DIG IGEN."
4600 PRINT "'DU MA UDFRI MIG!' GLAMMER HAN HÆST."
4610 PRINT "'JEG HOLDER DET IKKE UD MERE!'" :GOTO
      5020
4620 FOR I=1 TO 3:A(I)=INT(RND(1)*26)+65
4630 FOR J=0 TO I-1
4640 IF A(I)=A(J) THEN A(I)=INT(RND(1)*26)+65:GOTO
4630 4650 NEXT J
4660 FOR K=1 TO 3
4670 IF A(K)=65 OR A(K)=69 OR A(K)=73 OR A(K)=79 OR
      A(K)=85 OR A(K)=89 THEN OM=1
4680 IF OM=0 THEN 4620
4690 NEXT K
4700 M$=CHR$(A(I))+M$
4710 NEXT I
4720 FOR I=1 TO 3:B$(I)=MID$(M$,INT(RND(1)*3)+1,1)
4730 FOR J=1 TO I-1
4740 IF B$(I)=B$(J) THEN
      B$(I)=MID$(M$,INT(RND(1)*3)+1,1):GOTO 4730
4750 NEXT J:NEXT I

```

```

4760 FOR I=1 TO 3
4770 C$=B$(I)+C$
4780 NEXT
4790 IF LEN(C$)>3 THEN C$="":GOTO 4720
4800 IF C$=M$ THEN 4720
4810 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET:PRINT
      CHR$(147) *****
4815 PRINT "PLUDSELIG STAR DER EN UHYGGELIG MUMIE"
4820 PRINT "FORAN DIG. HAN ER VIKLET IND I HALVT"
4830 PRINT "FORRADNEDE BANDAGER, OG MAN KAN"
4840 PRINT "TYDELT LUGTE, AT DET ER ET PAR"
4850 PRINT "TUSIND AR SIDEN, HAN HAR BØRSTET"
4860 PRINT "TÆNDER. MED HÆS OG SKURRENDE STEMME"
4870 PRINT "HVISER HAN TIL DIG: 'JEG VAR"
4880 PRINT "FARAO TUT LI HUTS SLAVE,"
4890 PRINT "OG DA MIN HERRE DØDE, BLEV DET MIG"
4900 PRINT "PALAGT AT VOGTE DENNE PYRAMIDE TIL"
4910 PRINT "TIDERNES ENDE. KUN EEN TING KAN"
4920 PRINT "UDFRI MIG AF MIN HVILELØSE VANDRING"
4930 PRINT "RUNDT I DISSE GANGE: AT EN,"
4940 PRINT "JEG MØDER, SIGER MIT NAVN."
4950 PRINT "DET ER MIG TILLADT AT RØBE, AT DET"
4960 PRINT "ER PÅ TRE BOGSTAVER, OG AT BOGSTAVERNE"
4970 PRINT "ER '";C$;"'. SÆT DEM SAMMEN I"
4980 PRINT "DEN RIGTIGE RÆKKEFØLGE OG GØR OS BEGGE"
4990 PRINT "EN TJENESTE - FOR HVIS DU IKKE GØR DET,"
5000 PRINT "TAGER JEG DINE MAGNETSTEN OG DE ANDRE"
5010 PRINT "OG LÆGGER DEM ALLE PÅ NYE STEDER!"
5020 PRINT "BOGSTAVERNE ER '";C$;"'. SIG MIT NAVN!"
5030 INPUT NA$
5040 IF NA$<>M$ THEN 5100
5050 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET:PRINT
      CHR$(147) *****
5055 PRINT "'DET ER RIGTIGT!' JUBLER HAN"
5060 PRINT "OG GLIDER OVER I DEN EVIGE HVILE.":PRINT

```

```

5070 NA=2:MU=-1
5080 FOR I=1 TO 3000:NEXT I
5090 RETURN
5100 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET:PRINT
      CHR$(147) *****
5105 PRINT "'FORKERT!' UDBRYDER HAN VRISSENT."
5110 PRINT "'JEG FATTER IKKE, DET SKULLE VÆRE SA"
5120 PRINT "SVÆRT! SOM STRAF TAGER JEG ALLE STEN"
5130 PRINT "- OGSA DEM, DU HAR - OG LÆGGER DEM
5140 PRINT "ANDRE STEDER!' OG MED EN LUNGELØS
5150 PRINT "LATTER FORSVINDER HAN I MØRKET -
5160 PRINT "OG MED HAM DINE MAGNETSTEN.":NA=1:
      MU=-1:S=0
5170 FOR I=1 TO 8000:NEXT I
5180 FOR I=1 TO 10:S(I)=INT(RND(1)*99)+2
5190 FOR J=0 TO I-1
5200 IF S(I)=S(J) OR S(I)=LO OR S(I)=SK THEN
      S(I)=INT(RND(1)*99)+2:GOTO 5190
5210 NEXT J:NEXT I
5220 RETURN
5230 END

```

Lad os begynde gennemgangen med en variabelliste:

Indekserede variabler

A(): Bruges i anagram-rutinen.

L(): Indeholder data for »døre«.

G(): Ditto – til brug for scanning af »øerne«.

LO\$(): Lokalitetsbeskrivelser før »uddeling«.

L\$((): Lokalitetsbeskrivelser efter »uddeling«.

S(): Magnetstenenes positioner.

S\$: Beskrivelse af magnetsten.

V\$: Beskrivelse af verdenshjørner (eller »døre«).

Enkeltvariabler

A: Fortæller, om det er første gang, hjælpe-rutinen læses.

K: Fortæller, om man ønsker at gå ud af spillet.

GA: Lokalitetsvariabel under scanning af »øerne«.

LO: Lokalitetsvariabel.

MU: Mumiens position.

NA: Fortæller, om mumien har været der og om dens navn er blevet gættet.

P,Q: Koordinater til brug ved PRINTning af kortet.

SK: Skattens position.

S\$: CHR\$(13)

De øvrige variabler er lokale, og hensigten med dem burde fremgå af sammenhængen.

Gennemgang linie for linie (dog kun de mest interessante medtaget):

30: Vi hopper ned til indledningsteksten (se senere).

50: GA sættes – parallelt med LO – til 1. MU sættes til -1, hvil-

ket forhindrer mumien i at dukke op i utide på bl.a. LOkalitet 0 (se linierne 2040-2070). Linieskift får tildelt variabelen S\$.

90-100: DATA for verdenshjørner og magnetsten.

120-130: Grundtallet til brug for tilfældighedstalsgeneratoren findes ved hjælp af ur og dato. Hvis du har en Commodore eller lignende computer, der selv beregner dette grundtal, skal du springe disse to linier over. Skal du have et grundtal, men mangler TIME\$ og DATE\$ i din BASIC-version, så se afsnittet »Tilfældigheder« i kapitel 4.

150-190: Beregning af skattens og magnetstenenes positioner. J-løkken i linierne 160-190 sørger for, at der ikke bliver anbragt to sten på samme sted.

210-470: Første kortlægning af labyrinten (se evt. detaljeret gennemgang af processen i forrige kapitel).

490: Et lille trøstende ord til spilleren, der måske sidder og keder sig i det minuts tid, det tager computeren at kortlægge labyrinten.

500-900: Scanning af labyrinten. »Øerne« forbindes (se evt. gennemgangen af processen i forrige kapitel).

910: Endnu en lille trøstende bemærkning.

930-1410: Lokalitetsbeskrivelser. LO\$(1) vil (ifølge sagens natur) altid blive knyttet til Rum nr. 1, og LO\$(6) til skatkammeret. LO\$(3) vil blive påhæftet ethvert rum, der (kun) har udgang mod nord og syd eller øst og vest, og LO\$(4) enhver »gang«, der slår et knæk. LO\$(2) vil i første omgang blive hæftet på alle lokaliteter med 3 eller 4 udgange og siden hen sted for sted blive skiftet ud med LO\$(7)-LO\$(40) i det omfang, der er plads til dem.

1420-1550: Første »uddeling« af lokalitetsbeskrivelserne efter de

retningslinier, der er ridset op i kommentaren til linierne 930-1410.

1560: Endnu en lille trøstende bemærkning til spilleren, der stadig sidder og venter utålmodigt foran skærmen.

1570-1680: Her udskiftes de elementer i sættet $L\$()$, som har fået lokalitetsbeskrivelsen $LO\$(2)$ tildelt. Hvis der bliver nogen tilovers, har de altså stadig en lokalitetsbeskrivelse (»I et stort, tomt rum«), men ellers får de nye her. Hvis der på den anden side ikke er plads til alle de nye, stopper maskinen, når der ikke er flere rum med tre eller fire udgange tilovers. Dette kræver nok en nærmere forklaring, og den kommer her:

I linie 1570-1590 tæller vi ved hjælp af variablen LG, hvor mange rum med tre eller fire udgange, der egentlig er i spillet. I løkken i linie 1600-1680 fordeles lokalitetsbeskrivelserne $LO\$(7)$ - $LO\$(40)$ tilfældigt på disse rum. Der sørges i linie 1630 for, at ingen rum med andet end tre eller fire udgange kommer i betragtning, og i linie 1640-1650 sørges der for, at ingen lokalitetsbeskrivelse, som allerede er blevet brugt, bliver brugt en gang til.

1720-1730: Mumien får ordre til at dukke op i et ud af fyrre tilfælde, dersom mindst to af magnetstenene er blevet fundet.



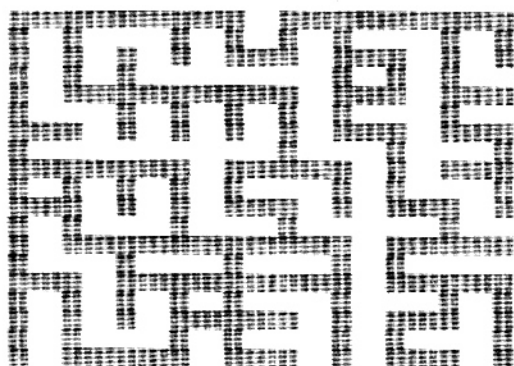
1910-1930: Vores input-rutine. Bemærk, at vi i dette spil benytter os af enkeltbogstavskommandoer (læs hjælpe-rutinen i linierne 3810-4310) og derfor kan bruge en INKEY\$-rutine (Commodore: GET) i stedet for at skulle trykke på retur-tasten, hver gang vi giver en kommando.

2040-2070: Vi snyder maskinen og opretter en »Lokalitet nr. Nul« for at give indtryk af, at vi godt kan forlade pyramiden. Se også linie 2130-2150.

2110-2220: Skærmtekst-rutine.

2240-3600: Korttegnings-rutine. Læs denne forklaring grundigt igennem, før du skriver programmet af. Hvis du har en datamat, hvis BASIC-dialekt adskiller sig blot en smule fra Microsoft- eller Commodore-BASIC, er dette et af de steder, hvor du højst sandsynligt vil løbe ind i problemer.

2260-2290: Her defineres de variabler, der skal bruges til at »tegne« kortet. I virkeligheden tegner vi det ikke, men PRINTER det, og til det bruger vi på IBM PC'en og beslægtede maskiner først og fremmest tegnet CHR\$(219) (der er et »negativt« eller »inverst« tomrum, som – for at bruge et billede – svarer til en sort firkant på et stykke hvidt papir) og nogle tomme mellemrum. Antallet af disse »sorte firkanter« og tomme mellemrum er i programmet beregnet ud fra en skærmbredde på 80, og til Commodoren er de tilsvarende »tegn« anført i REM-sætninger. Har du en anden maskine, må du selv finde ud af at erstatte dem. Det skulle ikke være så svært. Det hele går som sagt ud på at tegne en labyrint på skærmen – en labyrint, der fx kan komme til at se sådan ud:



2310-3440: Her PRINTes kortet ud på skærmen med »sorte firkanter« og tomme mellemrum.

3450-3600: Hvis du ikke har kommandoen LOCATE eller PRINT AT eller lignende i din BASIC-version (se indledningen af dette kapitel), må du springe disse linier over og nøjes med et kort uden angivelse af skatkammer og magnetsten. Du skal dog i så tilfælde beholde linierne 3540 samt 3580-3600.

I linie 3450-3470 definerer vi tegnene for henholdsvis din position, skattens position og magnetstenenes positioner med henblik på at anbringe dem på kortet. I Microsoft BASIC er CHR\$(176)-CHR\$(178) tre forskelligt rastede firkanter, men der er intet i vejen for, at du erstatter dem med andre tegn. Har du en Commodore, skal du rette dig efter, hvad der står i REM-sætningerne.

3480-3550: Vi går ned til den subrutine, der bestemmer præcis hvor på kortet de ovennævnte tegn skal PRINTes. Vi er nødt til her et øjeblik at erstatte LO med MI (noget skulle den jo hedde, ikke? MI står for Midlertidig), da maskinen skal bruge LO igen bagefter, og LO må derfor ikke forandres.

3560-3570: Her PRINTes diagramforklaringen, så spilleren kan se, hvilke tegn, der svarer til hvad.

3620-4340: Indledningen, hvoraf en del bliver genbrugt som hjælpe-rutine.

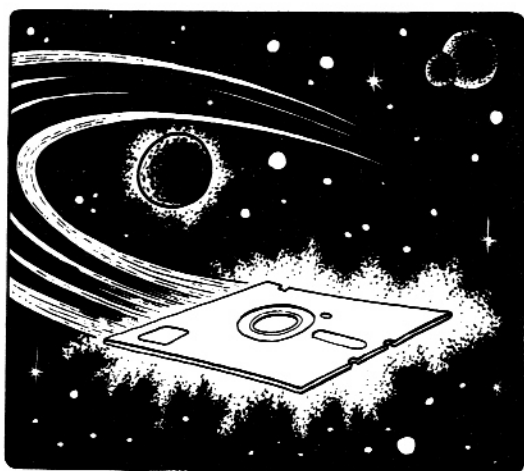
4350-4380: Mellemrumstangent-rutinen.

4400-4510: Udregning af de koordinater, der bestemmer, hvor på kortet din position, magnetstenenes positioner og skattens position skal PRINTes. Bemærk, at ligningen i linie 4500 er anderledes i Microsoft-versionen end i Commodore-versionen. Passer ingen af de to alternativer til din maskine, må du selv lægge hjernen i blød. P står for den linie, som den ting, der skal angives på kortet, er på, og Q for dens position på denne linie. Du kan sandsynligvis nå et brugbart resultat ved at beholde det, der står mellem parenteserne og eksperimentere med henholdsvis det første og det sidste tal i ligningen.

4530-4560: Magnetstens-rutinen.

4580-5220: Mumie-rutinen, som indbefatter anagram-rutinen, som du kan se detaljeret beskrevet i kapitel 4.

8 · Eventyrspil på disketter



Samtlige programmer, der er med i denne bog, kan spilles, dersom du blot har en kassettebåndstation. Der er ikke medtaget nogen, der *kun* kan spilles på computere med diskettestationer – men da diskettestationer åbner mulighed for at gøre spillene længere – praktisk taget så lange, man ønsker dem – skal jeg ikke undlade at give en lille forklaring på, hvordan man gør dette. Så hvis du ikke har mindst én diskettestation forbundet til din computer, kan du godt springe dette kapitel over og hoppe videre til det sidste, »Mordet i den lukkede boks«, der indeholder et detektivspil, som du sikkert kan få mange timer til at gå med.

Hvis du *har* mindst én diskettestation, kan du trygt læse videre – medmindre din BASIC-version mangler den kommando, der på Microsoft BASIC hedder CHAIN. Så vil det nemlig også være tidsspilde for dig at læse videre i dette kapitel.

Kommandoen CHAIN skal bruges, når man sammenkører et BASIC-program med et andet. Og det er netop ved at dele et

eventyrspil op i flere programmer (eller sektioner), at man kan gøre det så langt, man måtte ønske det. Det eneste, man skal sørge for, er at sektionerne hver for sig bliver korte nok til at kunne rummes i maskinens RAM.

Lad os tage et opdigtet eventyrspil (vi kalder det EVENTYR) på 100 lokaliteter, et eventyrspil, som (leger vi) vi har broderet så meget på, at det er blevet for langt til at kunne rummes i maskinens RAM. Lad os prøve at se på, hvordan vi deler det op i sektioner, så det kan være der.

Vores EVENTYR rummer en indledning, der kun skal køres én gang under spillet og altså bare står og tager plads op i hukommelsen, når den én gang er blevet PRINTet ud på skærmen og fjernet igen. Vi kunne altså vinde dyrebar hukommelsesplads ved at smide indledningen ud, når vi har brugt den. Vi laver et program, der kun indeholder indledningen (dvs. indlednings tekst plus dimensionering og indlæsning af variabler i DATA-sætninger). Vi kalder denne programsektion for EVENTYR.IND, og vi afslutter den med sætningen CHAIN »EVENTYR.LO«, 10,ALL.

Herved smides alt ud med undtagelse af variablerne, hvis indhold køres over i næste sektion, som automatisk bliver LOAD'et via CHAIN-kommandoen og RUNnet fra linie 10. Det vil ske hurtigt og effektivt, og spilleren vil sandsynligvis ikke lægge mærke til andet, end at diskettestationen »mumler« et øjeblik.

I sektionen EVENTYR.LO har vi anbragt vores lokalitetsbeskrivelser – og den eneste grund til, at vi behøver have dem i en sektion for sig selv, er at de skal kunne gennemlæses selvstændigt, hvis vi har GEMt SPillet og bagefter FORTsætter det. Du husker nok, at maskinen ikke vil tage lokalitetsbeskrivelserne med i sin GEMmefil, hvis de indeholder lineskift, så derfor er vi nødt til at have lokalitetsbeskrivelserne i deres egen sektion, og det er den, vi kører igennem nu. Dersom du har valgt en anden løsning end den med de indbyggede lineskift og anbragt dine lokalitetsbeskrivelser i DATA-sætninger, behøver du ikke denne sektion EVENTYR.LO. Så skal du blot huske at tage dem med i sektionen EVENTYR.IND og CHAINe denne (kæde den sammen) med næste sektion, EVENTYR.HOV (Eventyrets hovedprogram).

Det gør vi også nu, i slutningen af EVENTYR.LO, med sætningen CHAIN »EVENTYR.HOV«,10,ALL.

Og så videre og så videre. Hvis hovedprogrammet stadig skulle vise sig at være for langt, kan du pille afslutningen ud af det og lægge den i en sektion kaldet EVENTYR.AFS, eller du kan lægge hjælpe-rutinen eller gemme/fortsætte-rutinen i en sektion for sig. Det eneste, du skal huske, er at overføre variablerne fra den ene sektion til den anden, hver gang du skifter imellem dem. Og når du FORTsætter SPillet efter at have GEMt det, skal du huske at give maskinen ordre til at læse EVENTYR.LO igen - nem for også at få lokalitetsbeskrivelserne med (hvis du da ikke har dem i DATA-sætninger).

Du kan pille en hvilken som helst rutine ud af hovedprogrammet og lægge den ud på disketten for at formindske hovedprogrammet - men læg ikke flere end højst nødvendigt ud på denne måde. Husk, at det slider på diskettestationens læsehoved, hver gang det bliver brugt. Og husk, at det *tager* et øjeblik længere end normalt, så gør det kun med de rutiner, der ikke bliver brugt så tit. Det er fx irriterende at skulle sidde og vente 10 sekunder hver gang man giver maskinen besked på at GÅ et eller andet sted hen.

Hvis dit hovedprogram efter disse manøvrer stadig skulle vise sig at være for langt, er der intet i vejen for, at du deler det op i to, tre eller flere hovedprogrammer. Du kunne fx lave et hovedprogram, der omhandler lokalitet 1-25... og altså undlade at medtage de - sikkert mange - IF THEN-sætninger, der ikke knytter sig til disse lokaliteter. Dem tager du så til gengæld med i dit hovedprogram nr. 2, der fx kunne omhandle lokalitet 26-50 og ikke behøver indeholde de IF THEN-sætninger, der knytter sig til spillets øvrige lokaliteter. Og så videre og så videre.

Denne fremgangsmåde har den ulempe, at man er nødt til at have flere programmer med »overlapninger« - strengdelingsrutinen, gå-rutinen, tage-rutinen osv. skal være med i dem alle i måske næsten identiske versioner - men der er *meget* plads på en diskette... og hvis det ikke kan være på én, kan man bare tage flere i brug!

Laver du flere hovedprogrammer, der hver koncentrerer sig

om bestemte lokaliteter, er der ingen grund til at medtage et fuldt sæt lokalitetsbeskrivelser hele spillet igennem. De fylder nemlig godt op i hukommelsen, og man kan lige så godt smide dem ud, man ikke skal bruge i den pågældende sektion.

Hvis man har 100 lokaliteter, som man gerne vil køre i fire hovedprogrammer, kan man bruge følgende fremgangsmåde:

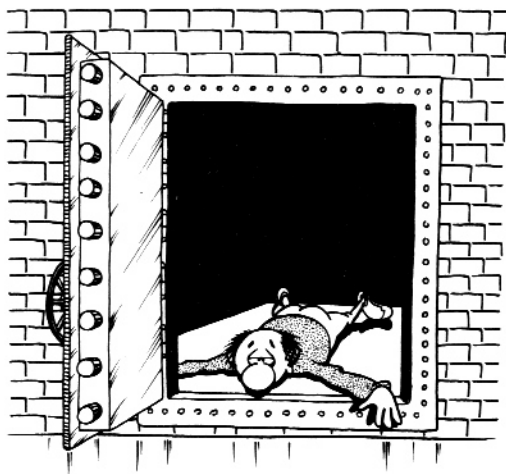
1. DIMensioner dem alle 100 i indledningssektionen.

2. Opret fire lokalitets-sektioner (1-25, 26-50, 51-75, 76-100) og indlæs hver af dem umiddelbart inden den hovedsektion, de hører til.

3. Smid dem, du ikke skal bruge, ud af hukommelsen ved at gøre dem »tomme«, dvs. lig med »«. På denne måde er de der stadig, så maskinen kan tælle dem, men de fylder ikke ret meget. Hvis du smed dem *helt* ud (fx Lokalitetsbeskrivelse nr. 1-25), ville maskinen tro, at Lokalitetsbeskrivelse nr. 26 var lig med nummer 1!

Mere vil jeg ikke skrive om dette. Jeg er sikker på, at du selv kan eksperimentere dig frem til en tilfredsstillende løsning ud fra de ovenstående retningslinier.

9 · Mordet i den lukkede boks



Som rosinen i pølseenden har jeg medtaget en lille gave til læserne: et spil, der går ud på at opklare et gådefuldt mord. Spilleren er detektiven, der har mulighed for at undersøge ting og lokaliteter, tage fingeraftryk og afhøre de mistænkte. Mere vil jeg ikke røbe. Jeg har ikke medtaget nogen forklaringer og har endda kodet en del af LISTningen, så du ikke kommer til at finde ud af for meget, når du skriver den af.

Du skal ved indskrivningen være opmærksom på én ting: De kodede sætninger (du vil nemt kunne genkende dem – de er det rene volapyk at se på) kan du skrive direkte af, dersom dine Æer, Øer og Åer har ASCII-værdierne 91, 92 og 93. Har de ikke det, skal du gøre følgende:

Hver gang du i en af disse kodede sætninger (se fx linierne 3770-3840) støder på tegnet »Ø«, skal du erstatte det med det tegn, der i din ASCII-tabel ligger lige efter Æ'et – det tegn, der har ASCII-værdien $\text{ASC}(»\text{Æ}«)+1$. På samme måde erstatter du

tegnet »Å« med det tegn, der har ASCII-værdien ASC(»Ø«)+1 – og tegnet »^« erstatter du med det tegn, der har ASCII-værdien ASC(»Å«)+1.

Har du hverken Æ, Ø eller Å på din computer, erstatter du tegnet »Ø« med »BD«, tegnet »Å« med »PD« (eller evt. med »1« – dersom du bruger et nul som dit »Ø«) og tegnet »^« med »BB«. *Men kun i de sætninger, der er kodet!* Koden er, som du sikkert allerede har regnet ud, ikke særlig svær at bryde – men hvis du gør det, finder du ud af, hvem morderen er ... og så er der ikke noget ved at spille spillet!

Her kommer LISTningen. God fornøjelse!

```
5 REM MORDET I DEN LUKKEDE BOKS *****
10 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
   CHR$(147) *****
20 PRINT "*****"
30 PRINT "      MORDET I DEN LUKKEDE BOKS"
40 PRINT:PRINT "ET EVENTYRSPIL FOR MESTERDETEKTIVER"
50 PRINT "*****"
60 PRINT:PRINT "HVIS DU HAR GEMT SPILLET":PRINT "OG
   GERNE VIL FORTSÆTTE"
70 PRINT "FRA DET STED, HVOR DU SLAP,":PRINT "SKAL
   DU BARE SKRIVE"
80 PRINT "'FORTSÆTTE SPIL'":PRINT "NAR DU FAR
   SPØRGSMålet"
90 PRINT "'HVAD VIL DU GØRE?'"
100 PRINT:PRINT "PA SAMME MADE KAN DU TIL ENHVER
   TID"
110 PRINT "GEMME SPILLET VED AT SKRIVE":PRINT "'GEM
   SPIL'":PRINT
120 PRINT "HVAD HEDDER DU";
130 INPUT NAS
140 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
   CHR$(147) *****
```



```

150 PRINT "OKAY, DU HEDDER ";NA$;",";PRINT "MEN NU
    FORESTILLER VI OS, AT"
160 PRINT "DU ER DEN BERØMTE KRIMINALKOMMISSÆR"
170 PRINT "NORTON STEELE FRA NEW YORK."
180 PRINT "EN DAG, HVOR DU SIDDER PÅ DIT KONTOR,"
190 PRINT "RINGER EN AF DINE ASSISTENTER"
200 PRINT "OG FORTÆLLER DIG, AT MAN HAR FUNDET"
210 PRINT "LIGET AF VEKSELEREREN JOHN EAGLE"
220 PRINT "I EN LUKKET BOKS I HANS FIRMA.":PRINT
    "DØREN TIL BOKSEN VAR LUKKET"
230 PRINT "INDEFRA, OG MAN BLEV NØDT TIL":PRINT "AT
    SPRÆNGE DØREN TIL BOKSEN"
240 PRINT "FOR AT KOMME DERIND. EAGLE HAVDE TRE"
250 PRINT "ANSATTE: PRIAM, SMITH OG RENFIELD."
260 PRINT "DE SIGER, DER ER FORSVUNDET":PRINT
    "DIAMANTER TIL EN VÆRDI AF"
270 PRINT "EN HALV MILLION DOLLARS FRA BOKSEN,"
280 PRINT "OG AT INGEN ANDRE HAR BESØGT":PRINT
    "FIRMAET DEN DAG.":PRINT
290 GOSUB 6270
300 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
    GET A$ *****
305 IF A$="" THEN 300
310 IF A$(">") THEN 300
320 PRINT:PRINT "ET ØJEBLIK!"
330 LO=2:AU=7:AN=30:GL=19:AL=24
340 DIM G$(AN),G$(AN),U$(AU),N$(AN)
350 DIM V$(4),L$(AL),L$(AL,4)
360 S$=CHR$(13)
370 T$="PÅ GADEN"+S$
380 U$="UDEN FOR "
390 V$=CHR$(14)
400 SA$=S$+"OG SENDER DE"
410 SB$=" HEN"+S$+"TIL TEKNISK AFDELING."
420 SN$=SA$+"N"+SB$

```

```

430 SM$=SA$+"M"+SB$
440 ST$=SA$+"T"+SB$
450 DT$="DU TAGER "
460 SA$=S$+"OG SENDER DE"
470 SB$=" HEN"+S$+"TIL TEKNISK AFDELING."
480 UN$="HAR JEG IKKE FAET"+S$+"TIL UNDERSØGELSE,"
    Siger HAN."
490 FOR I=1 TO GL:READ G(I):NEXT I
500 FOR I=1 TO GL:READ G$(I):NEXT I
510 FOR I=1 TO AU:READ U$(I):NEXT I
520 FOR I=1 TO AN:READ N$(I):NEXT I
530 FOR I=1 TO 4:READ V$(I):NEXT I
540 FOR I=1 TO AL:FOR J=1 TO 4:READ L(I,J):
    NEXT J:NEXT I
550 DATA 0,2,0,0,2,22,22,22,15,22,0,18,0,1,15,
    15,5,6,22
560 DATA EN RANSAGNINGSKENDELSE,DIN LOMMEBOG,ET
    PUSTERØR
570 DATA EN LILLE PIL TIL ET PUSTERØR,DIN
    ARBEJDSKASSE
580 DATA EN GRØN LÆNESTOL,ET LILLE UAFLAST SKAB AF
    TEAKTRÆ
590 DATA ET RYGEBOORD AF MAHOGNI,EN REGNEMASKINE,ET
    LIG,NOGLE KASSEBØGER
600 DATA NOGLE KRIMINALROMANER,EN NØGLE,RENFIELD -
    KONTORIST HOS EAGLE
610 DATA PRIAM - BOGHOLDER HOS EAGLE,SMITH -
    KONTORIST HOS EAGLE
620 DATA HUBBARD - CHEF FOR TEKNISK AFDELING
630 DATA WILSON - DOMMER,NOGLE PAPIRER
640 DATA GA,TAG,UND,GEM,FOR,SPØ,AFH
650 DATA RAN,LOM,PUS,PIL,ARB,LÆN,SKA,RYG,REG,LIG
660 DATA KAS,KRI,NØG,REN,PRI,SMI,HUB,WIL,PAP
670 DATA SPI,FIN,HUL,STE,VIN,DØR,DIA,NOR,SYD,ØST,VES
680 DATA NORDPA,SYDPA,ØSTPA,VESTPA

```

690 L\$(1)="I RENFIELDS HJEM."
 700 DATA 0,3,0,0
 710 L\$(2)="PA DIT KONTOR."
 720 DATA 0,4,0,0
 730 L\$(3)="FORAN RENFIELDS HOVEDDØR."
 740 DATA 1,10,0,0
 750 L\$(4)="I FORHALLEN"+S\$+"PA POLITISTATIONEN."
 760 DATA 2,11,5,0
 770 L\$(5)="I TEKNISK AFDELING."
 780 DATA 0,0,0,4
 790 L\$(6)="PA DOMMER WILSONS KONTOR"
 800 DATA 0,13,0,0
 810 L\$(7)=T\$+U\$+"SMITHS HJEM."
 820 DATA 0,14,8,0
 830 L\$(8)="FORAN SMITHS HOVEDDØR."
 840 DATA 0,0,0,7
 850 L\$(9)="I SMITHS HJEM."
 860 DATA 0,0,0,8
 870 L\$(10)=T\$+U\$+"RENFIELDS HJEM."
 880 DATA 3,0,11,0
 890 L\$(11)=T\$+U\$+"POLITISTATIONEN."
 900 DATA 4,0,12,10
 910 L\$(12)=T\$+U\$+"VEKSELERER-FIRMAET."
 920 DATA 0,15,13,11
 930 L\$(13)=T\$+U\$+"DOMHUSET."
 940 DATA 6,0,14,12
 950 L\$(14)=T\$+U\$+"PRIAMS HJEM."
 960 DATA 7,16,0,13
 970 L\$(15)="I VEKSELERER-FIRMAETS KONTOR."
 980 DATA 12,19,0,0
 990 L\$(16)="FORAN PRIAMS HOVEDDØR."
 1000 DATA 14,0,0,0
 1010 L\$(17)="I PRIAMS HJEM."
 1020 DATA 0,0,0,16
 1030 L\$(18)="VED ET KNEK I GANGEN."

```

1040 DATA 0,21,19,0
1050 L$(19)="I EN L-FORMET GANG"+S$+"BAG
      VEKSELERER-KONTORET. "
1060 DATA 15,22,20,18
1070 L$(20)="I DEN ØSTLIGE ENDE AF GANGEN. "
1080 DATA 0,0,0,19
1090 L$(21)="MIDT I DEN DEL AF GANGEN,"+S$+"DER
      LØBER NORD/SYD. "
1100 DATA 18,23,0,0
1110 L$(22)="INDE I"+S$+"VEKSELERER-FIRMAETS
      BOKSRUM. "
1120 DATA 19,0,0,0
1130 L$(23)="I DEN"+S$+"SYDLIGE ENDE AF GANGEN. "
1140 DATA 21,0,0,0
1150 L$(24)=U$+S$+"VEKSELERER-FIRMAETS BAGDØR. "
1160 DATA 23,0,0,0
1170 GOSUB 5940
1180 IF G(13)=-1 THEN L(23,2)=24
1190 IF PI=1 THEN L(16,3)=17
1200 IF SI=1 THEN L(8,3)=9
1210 PRINT:PRINT "HVAD VIL DU GØRE";
1220 INPUT B$
1230 TG=0:PRINT
1240 IF B$="R" THEN 1170
1250 IF B$="N" THEN B$="GA NORDPA"
1260 IF B$="S" THEN B$="GA SYDPA"
1270 IF B$="Ø" THEN B$="GA ØSTPA"
1280 IF B$="V" THEN B$="GA VESTPA"
1290 IF B$="STOP" THEN END
1300 FOR I=1 TO LEN(B$)
1310 IF MID$(B$,I,1)=" " THEN 1340
1320 NEXT I
1330 GOTO 6260
1340 UQ$=LEFT$(B$,I-1)
1350 NQ$=RIGHT$(B$, (LEN(B$)-I))

```

```

1360 IF UO$="GA" THEN 1410:REM IKKE HVIS A=AA *****
1370 IF LEN(UO$)<3 THEN 6260
1380 UO$=LEFT$(UO$,3):FOR I=1 TO AU:IF UO$=U$(I)
    THEN 1420
1390 NEXT I
1400 UO=0:GOTO 6260
1410 UO=1:GOTO 1430:REM IKKE HVIS A=AA *****
1420 UO=I
1430 IF LEN(NO$)<3 THEN 6260
1440 NO$=LEFT$(NO$,3):FOR I=1 TO AN:IF NO$=N$(I)
    THEN 1500
1450 NEXT I
1460 IF NO$="DOM" THEN NO=18:GOTO 1510
1470 IF NO$="STO" THEN NO=6:GOTO 1510
1480 IF NO$="BOR" THEN NO=8:GOTO 1510
1490 NO=0:GOTO 6260
1500 NO=I
1510 ON UO GOTO 1520,1560,2100,2880,3050,3170,6080
1520 IF NO<27 THEN 6260
1530 NO=NO-26
1540 IF L(LO,NO)=0 THEN PRINT "DU KAN IKKE GA DEN
    VEJ.":GOTO 1210
1550 LO=L(LO,NO):GOTO 1170
1560 IF G(NO)=-1 THEN 1910
1570 IF NO=21 THEN 1920
1580 IF NO>19 THEN 6260
1590 IF G(NO)<>LO THEN PRINT "ER IKKE HER.":GOTO
    1210
1600 IF LO<>5 THEN 1660
1610 IF NO=4 THEN 1640
1620 PRINT "DET ER DER INGEN GRUND TIL.":PRINT
    "HVORFOR SKULLE DU DOG"
1630 PRINT "SLÆBE RUNDT PA ":PRINT G$(NO);"?":GOTO
    1210

```

```

1640 PRINT "DU STIKKER DIG PÅ PILEN":PRINT "OG
      FALDER DØD OM."
1650 PRINT "BEDRE HELD NÆSTE GANG.":END
1660 IF NO>9 THEN 1680
1670 ON NO GOSUB 1730,1730,1690,1690,1730,1690,1690,
      1690,1690:GOTO 1210
1680 ON NO-9 GOSUB 1690,1690,1690,1730,1900,1900,
      1900,1900,1900,1690:GOTO 1210
1690 G(NO)=5:PRINT DT$:
1700 IF NO>9 THEN 1720
1710 ON NO GOSUB 0,0,1750,1760,0,1770,1780,1810,
      1820:RETURN
1720 ON NO-9 GOSUB 1830,1860,1870,0,0,0,0,0,0,1890:
      RETURN
1730 IF NO=2 THEN AP=1
1740 G(NO)=-1:PRINT "UDMÆRKET.":RETURN
1750 PRINT "PUSTERØRET";ST$:RETURN
1760 PRINT "PILEN";SN$:RETURN
1770 PRINT "LÆNESTOLEN";SN$:RETURN
1780 PRINT DT$;"SKABET";ST$
1790 IF G(11)=0 THEN G(11)=5
1800 RETURN
1810 PRINT "RYGEBORDET";ST$:RETURN
1820 PRINT "REGNEMASKINEN";SN$:RETURN
1830 PRINT "LIGET";ST$
1840 IF G(4)=0 THEN G(4)=5
1850 RETURN
1860 PRINT "KASSEBØGERNE";SM$:RETURN
1870 PRINT "KRIMINALROMANERNE";SM$:PRINT
1880 PRINT "I REOLEN BAG VED DEM":PRINT "FINDER DU
      ET PUSTERØR.":G(3)=LO:RETURN
1890 PRINT "PAPIRERNE";SM$:RETURN
1900 PRINT "VÆR NU LIDT ALVORLIG.":RETURN
1910 PRINT "DET HAR DU ALLEREDE GJORT.":GOTO 1210

```

```

1920 IF G(5)<>-1 THEN PRINT "DU HAR IKKE DIN
      ARBEJDESTASKE.":GOTO 1210
1930 PRINT "PA HVEM ELLER HVAD";
1940 INPUT C$
1950 PRINT:IF LEN(C$)<3 THEN PRINT "OM IGEN!":GOTO
1930 1960 C$=LEFT$(C$,3)
1970 FOR I=1 TO GL:IF C$<>N$(I) THEN AA=0:
      NEXT I:GOTO 2060
1980 IF C$=N$(3) AND G(3)=LO THEN PU=1
1990 IF C$=N$(16) AND G(16)=LO THEN SM=1
2000 IF (C$=N$(18) AND G(18)=LO) OR (C$=N$(17) AND
      G(17)=LO) THEN 2020
2010 GOTO 2030
2020 PRINT "DU ER POKKERS MISTÆNKSOM, HVA'?"
2030 FOR I=1 TO AN
2040 IF C$=N$(I) AND (G(I)=LO OR G(I)=-1) THEN AA=1
2050 NEXT I
2060 IF AA=1 THEN FI=1:GOTO 2080
2070 AA=0:GOTO 6260
2080 PRINT DT$;"FINGERAFTRYKKENE";SM$
2090 GOTO 1210
2100 IF G(NO)=-1 OR (G(1)=0 AND (PI=1 OR SI=1)) THEN
      2140
2110 IF NO>25 THEN 6260
2120 IF NO>19 THEN 2140
2130 IF G(NO)<>LO THEN PRINT "FINDES IKKE HER.":
      GOTO 1210
2140 IF NO>8 AND NO<17 THEN 2170
2150 IF NO>16 THEN 2180
2160 ON NO GOSUB 2190,2210,2420,2430,2440,2450,2460,
      2490:GOTO 1210
2170 ON NO-8 GOSUB 2510,2520,2560,2570,2580,2770,
      2770,2770:GOTO 1210
2180 ON NO-16 GOSUB

```

```

2770,2770,2590,2770,2800,2820,2600,
    2780,2850:GOTO 1210
2190 PRINT "DEN GIVER DIG RET":PRINT "TIL AT
    RANSAGE"
2200 PRINT "DEN MISTÆNKTES HJEM.":RETURN
2210 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****
2220 PRINT "I LOMMEBOGEN STAR.":PRINT:PRINT "DU KAN
    BRUGE FØLGENDE"
2230 PRINT "KOMMANDOER I DETTE SPIL.":PRINT "GA -
    TAGE - UNDERSØGE - SPØRGE"
2240 PRINT "SAMT AFHØRE"
2250 PRINT "(EFTERFULGT AF ET NAVNEORD":PRINT "ELLER
    VERDENSHJØRNE - F.EKS":AP=1
2260 PRINT "'UNDERSØGE STED' ELLER":PRINT "'GA
    NORDPA'.":PRINT
2270 PRINT "DU KAN NØJES MED AT SKRIVE"
2280 PRINT "'N' FOR 'GA NORDPA'"
2290 PRINT "OG 'S' FOR 'GA SYDPA' OSV.":PRINT
2300 PRINT:PRINT "HVIS DU ØNSKER AT RYDDE SKÆRMEN"
2310 PRINT "FOR AT SE, HVOR DU ER,"
2320 PRINT "SA SKRIV 'R'.":PRINT
2330 GOSUB 6270
2340 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV I STEDET:
    GET A$ *****
2345 IF A$="" THEN 2340
2350 IF A$(">") THEN 2340
2360 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
    CHR$(147) *****
2365 PRINT "NAR DU HAR EN MISTÆNKT,":PRINT "KAN DU
    SPØRGE DOMMEREN"
2370 PRINT "OM EN RANSAGNINGSKENDELSE.":PRINT
2380 PRINT "DER ER KUN EEN MORDER,":PRINT "OG HAN
    HAR VÆRET ALENE"

```



```

2390 PRINT "OM SIN MISGERNING.":PRINT:PRINT "GOD
      JAGT!":PRINT
2400 PRINT "OBS! HVIS DU ØNSKER AT STOPPE SPILLET,"
2410 PRINT "SKAL DU BARE SKRIVE 'STOP'!":PRINT:
      RETURN
2420 PRINT "DET LUGTER AF TOBAK":PRINT "OG DER ER
      FINGERAFTRYK PÅ DET.":RETURN
2430 PRINT "DEN LUGTER AF TOBAK.":RETURN
2440 PRINT "DEN INDEHOLDER BL.A.":PRINT "DIT
      FINGERAFTRYKSUDSTYR.":RETURN
2450 PRINT "DEN ER AF PLYS":PRINT "OG TEMMELIG
      LUVSLIDT.":RETURN
2460 IF G(11)<>0 THEN PRINT "DET ER TOMT.":RETURN
2470 PRINT "DET INDEHOLDER":PRINT "NOGLE
      KASSEBØGER."
2480 PRINT "DU TAGER DEM UD.":G(11)=LO:RETURN
2490 IF LO=22 THEN PRINT "DET ER VÆLTET.":RETURN
2500 PRINT "DER ER IKKE NOGET":PRINT "USÆDVANLIGT
      VED DET.":RETURN
2510 PRINT "DER ER INTET":PRINT "USÆDVANLIGT VED
      DEN.":RETURN
2520 IF G(4)<>0 THEN 2550
2530 PRINT "DER SIDDER EN LILLE PIL":PRINT "FRA ET
      PUSTERØR I RYGGEN PÅ LIGET."
2540 PRINT "DU TAGER DEN UD.":G(4)=LO:RETURN
2550 PRINT "DU FAR IKKE SELV NOGET":PRINT "UD AF AT
      UNDERSØGE DET.":RETURN
2560 PRINT "DE INDEHOLDER REGNSKABER.":RETURN
2570 PRINT "DE ER AF JOHN DICKSON CARR.":RETURN
2580 PRINT "DET ER EN GANSKE":PRINT "ALMINDELIG
      YALENØGLE.":RETURN
2590 PRINT "DER STAR EN MASSE TAL PÅ DEM.":RETURN
2600 PRINT "DU SER:"
2610 IN=0

```

```

2620 FOR I=1 TO AN:IF G(I)=LO THEN PRINT G$(I):IN=1
2630 NEXT I
2640 IF LO>18 AND LO<24 THEN IN=1
2650 IF LO=20 THEN PRINT "ET VINDUE"
2660 IF LO=23 THEN PRINT "EN DØR"
2670 IF LO=22 THEN PRINT "ET HUL I VESTVÆGGEN."
2680 IF LO=21 THEN PRINT "ET HUL I ØSTVÆGGEN."
2690 IF LO=19 OR LO=22 THEN PRINT "EN SPRÆNGT DØR."
2700 IF LO<>22 THEN 2730
2710 PRINT:PRINT "MURENE BESTÅR AF EN HALV METER"
2720 PRINT "TYKT ARMERET BETON, OG DEN":PRINT
      "ENESTE VEJ IND ER GENNEM DØREN."
2730 IF IN=0 THEN PRINT "INTET AF INTERESSE."
2740 IF G(1)=-1 AND LO=1 THEN IN=2
2750 IF IN=2 THEN 6040
2760 RETURN
2770 PRINT "DET BLIVER DU IKKE KLOGERE AF.":RETURN
2780 IF LO<>20 THEN 2770
2790 PRINT "DET ER FORSYNET MED TREMMER":PRINT "OG
      LAST INDEFRA.":RETURN
2800 PRINT "DET ER ET JOB":PRINT "FOR TEKNISK
      AFDELING."
2810 PRINT "DU KAN KUN TAGE DEM":PRINT "OG SENDE DEM
      VIDERE.":RETURN
2820 IF LO<>22 AND LO<>21 THEN 2770
2830 PRINT "DET ER CA. 5 GANGE 5 CM"
2840 PRINT "OG GÅR HELE VEJEN GENNEM VÆGGEN.":RETURN
2850 IF (LO=23 AND G(13)=0) OR (LO=8 AND SI=0) OR
      (LO=16 AND PI=0) THEN 2870
2860 PRINT "DEN ER ÅBEN.":RETURN
2870 PRINT "DEN ER LAST.":RETURN
2880 IF NO=20 THEN 2900
2890 PRINT "DET ENESTE, DU KAN GEMME,":PRINT "ER
      SPILLET.":GOTO 1210

```

```

2900 OPEN "O",1,"BO-DATA":REM COMMODORE-BRUGER,
      SKRIV I STEDET: OPEN 1,1,1,"BO-DATA" *****
2910 PRINT# 1,I1
2911 PRINT# 1,I4
2912 PRINT# 1,S1
2913 PRINT# 1,P1
2914 PRINT# 1,F1
2915 PRINT# 1,PU
2916 PRINT# 1,SM
2917 PRINT# 1,AP
2918 PRINT# 1,F1
2919 PRINT# 1,F2
2920 PRINT# 1,F3
2921 PRINT# 1,LO
2922 PRINT# 1,NA$
2928 FOR I=1 TO GL:PRINT# 1,G(I):NEXT I
2930 FOR I=1 TO GL:PRINT# 1,G$(I):NEXT I
2940 FOR I=1 TO AU:PRINT# 1,U$(I):NEXT I
2950 FOR I=1 TO AN:PRINT# 1,N$(I):NEXT I
2960 FOR I=1 TO 4:PRINT# 1,V$(I):NEXT I
2970 FOR I=1 TO AL:FOR J=1 TO 4:PRINT# 1,L(I,J):NEXT
      J:NEXT I
2980 CLOSE 1
2990 PRINT "ØNSKER DU AT FORTSÆTTE SPILLET?"
3000 PRINT:PRINT "TRYK PÅ 'J' FOR 'JA':PRINT "ELLER
      'N' FOR 'NEJ'."
3010 A$=INKEY$:REM COMMODORE-BRUGER, SKRIV OM TIL:
      GET A$ *****
3015 IF A$="" THEN 3010
3020 IF A$="J" THEN 1170
3030 IF A$="N" THEN END
3040 GOTO 3010
3050 IF NO=20 THEN 3070
3060 PRINT "DET ENESTE,":PRINT "DU KAN FORTSÆTTE, ER
      SPILLET.":GOTO 1210

```

```

3070 OPEN "I",1,"BO-DATA":REM COMMODORE-BRUGER,
      SKRIV OM TIL: OPEN 1,1,0,"BO-DATA" *****
3090 INPUT# 1,I1
3091 INPUT# 1,I4
3092 INPUT# 1,SI
3093 INPUT# 1,PI
3094 INPUT# 1,FI
3095 INPUT# 1,PU
3096 INPUT# 1,SM
3097 INPUT# 1,AP
3098 INPUT# 1,F1
3099 INPUT# 1,F2
3100 INPUT# 1,F3
3101 INPUT# 1,LO
3102 INPUT# 1,NA$
3108 FOR I=1 TO GL:INPUT# 1,G(I):NEXT I
3110 FOR I=1 TO GL:INPUT# 1,G$(I):NEXT I
3120 FOR I=1 TO AU:INPUT# 1,U$(I):NEXT I
3130 FOR I=1 TO AN:INPUT# 1,N$(I):NEXT I
3140 FOR I=1 TO 4:INPUT# 1,V$(I):NEXT I
3150 FOR I=1 TO AL:FOR J=1 TO 4:INPUT# 1,L(I,J):NEXT
      J:NEXT I
3160 CLOSE 1:GOTO 1170
3170 IF NO<14 OR NO>18 THEN PRINT "VÆR NU IKKE
      FJOLLET.":GOTO 1210
3180 IF G(NO)<>LO THEN PRINT "FINDES IKKE HER.":GOTO
      1210
3190 PRINT "OM HVAD ELLER HVEM";
3200 INPUT D1$:PRINT
3210 IF LEN(D1$)<3 THEN PRINT "OM IGEN!":PRINT:GOTO
      3190
3220 D$=LEFT$(D1$,3)
3230 IF D$="STO" THEN D$=N$(6)
3240 FOR I=1 TO AN:IF D$=N$(I) THEN TG=I:GOTO 3270
3250 NEXT I

```

```

3260 GOTO 5860
3270 IF TG>10 AND TG<21 THEN 3300
3280 IF TG>20 THEN 3310
3290 ON TG GOTO 3320,4020,4040,4270,4310,4320,4390,
      4470,4520,4600:GOTO 1210
3300 ON TG-10 GOTO 4670,4800,4860,4960,5350,5360,
      5370,5370,5410,5640:GOTO 1210
3310 ON TG-20 GOTO 5500,5630,5670,5700,5760,5600,
      5840,5840,5840,5840:GOTO 1210
3320 ON NO-13 GOTO 4000,4000,4000,3330,3340
3330 PRINT "'DET ER IKKE MIT BORD,'" :PRINT "SIGER
      HUBBARD.":GOTO 1210
3340 PRINT "'HVEM MISTENKER DE":PRINT "FOR MORDET?'
      SPØRGER WILSON.":PRINT
3350 PRINT "JEG MISTENKER:"
3360 INPUT E$
3370 PRINT:E$=LEFT$(E$,3)
3380 IF E$=N$(14) OR E$=N$(15) OR E$=N$(16) THEN
      3400
3390 PRINT "'NU VASER DE,' SIGER WILSON.":GOTO 1210
3400 IF F1=0 OR F2=0 OR F3=0 THEN 3420
3410 GOTO 3450
3420 PRINT "'DE HAR IKKE ENGANG GJORT":PRINT "DEM
      DEN ULEJLIGHED AT"

```



PRIAM



SMITH



EAGLE

```

3430 PRINT "AFHØRE SAMTLIGE MISTÆNKTE,":PRINT "OG SA
      TILLADER DE DEM"
3440 PRINT "AT UDPEGE EN MORDER!":GOTO 3640
3450 IF E*(N*(16) THEN 3580
3460 IF SI(>)1 THEN 3510
3470 PRINT "'DE HAR ALLEREDE FAET LOV TIL":PRINT "AT
      RANSAGE HANS HJEM. DE MA"
3480 PRINT "FINDE DE STJALNE DIAMANTER":PRINT "OG
      BEVISE, AT HAN HAR TAGET DEM,"
3490 PRINT "HVIS DE VIL SIGTE HAM FOR MORD,'"
3500 PRINT "SIGER DOMMER WILSON.":GOTO 1210
3510 IF I1(>)1 THEN 3620
3520 Y*="(IBOT!GJOHFSBGUSZL!Q^!QVTUFSASFU":GOSUB
      6110:SI=1:G(16)=9
3530 PRINT "ER ET MEGET ALVORLIGT INDICIUM":PRINT
      "IMOD HAM. DE SKAL FA LOV"
3540 PRINT "TIL AT RANSAGE HANS HJEM.":PRINT "MEN
      HVIS DE IKKE FINDER"
3550 PRINT "DE STJALNE DIAMANTER, VIL EN":PRINT "SAG
      IMOD HAM IKKE KUNNE"
3560 PRINT "HOLDE I RETTEN,' SIGER WILSON"
3570 PRINT "OG GIVER DIG EN RANSAGNINGSKENDELSE.":
      G(1)=-1:GOTO 1210
3580 IF E*(N*(15) THEN 3620
3590 IF PI=1 THEN 3470
3600 IF I4=0 THEN 3620
3610 Y*="(IBOT!VOEFSTMØC":GOSUB 6110:PI=1:G(15)=17:
      GOTO 3530
3620 IF G(4)<>0 THEN 3670
3630 PRINT "'VI VED IKKE ENGANG, HVILKET":PRINT
      "VABEN, DER ER BRUGT."
3640 PRINT "JEG KAN IKKE GIVE DEM LOV":PRINT "TIL AT
      RANSAGE NOGENS HJEM"
3650 PRINT "PA SA SPINKELT ET GRUNDLAG.":PRINT "DE
      MA FORTSÆTTE DERES"

```

```

3660 PRINT "UNDERSØGELSER," SIGER HAN.":GOTO 1210
3670 X$="NFE!FO!GPSHJGUFU!QJM-"+V$+
      "NFO!IWPSEBO!CMFW!QJMF!BOCSBHU"
3680 Z$="J!SZHHFO!Q^!IBNQ(!TQASHFS!XJMTPQ/"
→ 3690 Y$="(WJ!WFE-!BU!FBHMF!CMFW!ESØCU":GOSUB 6110:
      X$="":Z$="":PRINT
3700 PRINT "VED HJÆLP AF:"
3710 INPUT F$
3720 PRINT:F$=LEFT$(F$,3)
3730 IF F$="STO" THEN F$=N$(6)
3740 IF F$(>)N$(3) THEN 3830
3750 IF G(3)<>0 THEN 3780
3760 PRINT "'SA VIL JEG RADE DEM TIL":PRINT "AT
      FINDE";
3770 Y$="!EFUUF!QVTUFSAS/":GOSUB 6110:GOTO 3640
3780 PRINT "'OG EFTERSOM BOKSEN VAR LAST":PRINT
      "INDEFRA, VIL DE VEL PASTA,"
3790 Y$="BU!NPSEFSFO!TLAE!QJMF!JOE"+V$+
      "J!SZHHFO!Q^!FBHMF!HFOOFN"
3800 X$="USØLLBOBMFOQ"+V$+
      "EFSFT!NJTUØOLUF!IBS!CBSF!FU":GOSUB 6110:X$=""
3810 PRINT "VANDTÆT ALIBI. INGEN HAR "
3820 PRINT "BEFUNDET SIG ALENE I GANGEN.":GOTO 3640
3830 IF F$(>)N$(6) THEN 3990
3840 IF E$(>)N$(14) THEN 3950
3850 Y1$=V$+"IBS!LVØOFU!LPNNF!VE!J!HBOHFO/"+V$+
      "IBO!BOCSBHUF!QJMF!J!TUPMFO-"
3860 Y2$=V$+"PH!EB!FBHMF!TBUUF!TJH!J!EFO-!GJL"
3870 Y3$="PH!EFU!BU!JOHFO"
3880 Y$="(EFU!WJMMF!GPSLMBSF!C^EF!EFU"+V$+
      "WØMUFEF!CPSE!"+Y3$+Y1$+Y2$
3890 X1$="PH!WØMUFEF!CPSEFU-!GAS!IBO"+V$+
      "TUZSUFEF!EAE!PN/!SFOGJFME!WBS"+V$
3900 X2$="BMFOF!J!CPLTFO-!NFOT!FBHMF!HJL"+V$

```

```

3910 X3$="JOE!PH!CBE!Q$JBN!PN!SFHOFNBTLJQFO~"+
    V$+"PH!IBO!LVOOF!IBWF!UBHFU"
3920 X$="IBO!EFO!J!SZHHFO~!SFKTUF!TJH"+V$+X1$+
    X2$+X3$:G(1)=-1
3930 Z$="EJBNBOUFSOF!J!TJO!UBTLF/!EFUUF"
3940 GOSUB 6110:X$="":Z$="":GOTO 3530
3950 Y1$="NFO!EFSFT!NJTUØOLUF!WBS"+V$+
    "JLLF!Q^!OPHFU!UJETQVOLU"+V$
3960 Y2$="BMFOF!J!CPLTFQ!PH!IBS!EFSGPS"+V$+
    "JLLF!LVOOFU!QJMMF!WFE!TUPMFO"
3970 Y3$= V$+"FMMFS!TUKØMF!EJBNBOUFSOF/"
3980 Y$="(ZEFSTU!JOUFSFTTBOU!UFPSJ~"+V$+Y1$+
    Y2$+Y3$:GOSUB 6110:GOTO 3640
3990 PRINT "'NU VRØVLER DE,' SIGER WILSON.":GOTO
    1210
4000 PRINT "'SADAN EN SKAL DE DA VIST":PRINT "HAVE
    HOS EN DOMMER,'"
4010 PRINT "SIGER HAN.":GOTO 1210
4020 PRINT "'HVAD SKULLE JEG VIDE OM DEN?'"
4030 PRINT "SPØRGER HAN - MED RETTE.":GOTO 1210
4040 ON NO-13 GOTO 4260,4260,4180,4050,4230
4050 IF G(3)<>5 AND PU=0 THEN PRINT "'DET ";UN$:GOTO
    1210
4060 IF PU<>0 THEN 4100
4070 PRINT "'DEN MAND, DE SENDTE HERHEN MED DET,"
4080 Y$="LPN!UJM!BU!UWØSF!GJOHFSBGUSZLLFOF!VE~(":
    GOSUB 6110
4090 PRINT "SIGER HAN. 'DE SKULLE HAVE":PRINT "TAGET
    DEM PA STEDET.'":GOTO 1210
4100 IF SM<>1 THEN 4140
4110 Y$="(EFU!FS!TNJUIT!GJOHFSBGUSZL~"
4120 X$="EFS!TJEEFS!Q^!QVTUFSASFU~("+V$+"TJHFS!IBO/"
4130 GOSUB 6110:X$="":I1=1:GOTO 1210
4140 PRINT "'VI HAR IKKE KUNNET IDENTIFICERE"
4150 Y$="GJOHFSBGUSZLLFOF!Q^!QVTUFSASFU":GOSUB 6110

```




```
4160 PRINT "UD FRA DET MATERIALE, DE HAR"  
4170 PRINT "GIVET OS,' SIGER HAN.":GOTO 1210  
4180 IF I1=0 THEN 4260  
4190 PRINT "'JEG KONFISKEREDE DET FRA EN DRENG,"  
4200 PRINT "DER GENEREDE MIG PA GADEN,'" :PRINT  
    "SIGER HAN. 'JEG HAR HAFT DET"  
4210 PRINT "LIGGENDE I MIN SKUFFE":PRINT "I TO UGER.  
    NOGEN MA HAVE"  
4220 PRINT "FJERNET DET UDEN MIT VIDENDE.'" :GOTO  
    1210  
4230 PRINT "'DET ER DEM, DER SKAL DRAGE":PRINT  
    "KONKLUSIONERNE,' SIGER HAN."  
4240 PRINT "'DE KAN SPØRGE, OM JEG VIL GIVE":PRINT  
    "DEM EN RANSAGNINGSKENDELSE,"  
4250 PRINT "HVIS DE MENER, DE HAR":PRINT "BEVISER  
    MOD NOGEN.'" :GOTO 1210  
4260 PRINT "'KENDER JEG IKKE NOGET TIL,'" :PRINT  
    "SIGER HAN.":GOTO 1210
```

```

4270 ON NO-13 GOTO 4260,4260,4260,4280,4230
4280 IF G(4)<>5 THEN PRINT "'DEN ";UN$:GOTO 1210
4290 Y$="(EFS!WBS!LPDFOUSFSFU-"+V$+
      "EAEC SJOHFOEF!OJLPUJO":GOSUB 6110
4300 PRINT "PA DEN,' SIGER HAN.":GOTO 1210
4310 GOTO 4260
4320 ON NO-13 GOTO 4370,4370,4370,4330,4230
4330 IF G(6)<>5 THEN PRINT "'DEN ";UN$:GOTO 1210
4340 Y$="(EFS!WBS!GMFSF!TN^!SFWOFS"
4350 X$="J!SZHHFO!Q^!EFO-(":GOSUB 6110:X$=""
4360 PRINT "SIGER HAN.":GOTO 1210
4370 PRINT "'MR. EAGLE SAD ALTID I DEN,"
4380 PRINT "NAR HAN ARBEJDEDE INDE":PRINT "I
      BOKSEN,' SIGER HAN.":GOTO 1210
4390 ON NO-13 GOTO 4440,4440,4440,4400,4230
4400 IF G(7)<>5 THEN PRINT "'DET ";UN$:GOTO 1210
4410 PRINT "'DER ER IKKE NOGET"
4420 PRINT "USÆDVANLIGT VED DET,' SIGER HAN."
4430 PRINT "'EAGLE OPBEVAREDE SINE":PRINT
      "KASSEBØGER I DET.':GOTO 1210
4440 PRINT "'DA JEG SIDST SA EAGLE I LIVE,"
4450 PRINT "LA DER DIAMANTER TIL EN VÆRDI":PRINT "AF
      EN HALV MILLION DOLLARS"
4460 PRINT "I TEAKTRÆSSKABET,' SIGER HAN.":GOTO 1210
4470 ON NO-13 GOTO 4500,4500,4500,4480,4230
4480 IF G(8)<>5 THEN PRINT "'DET ";UN$:GOTO 1210
4490 GOTO 4410
4500 PRINT "'EAGLE HAVDE SINE PAPIRER PA DET,":PRINT
      "NAR HAN ARBEJDEDE INDE"
4510 PRINT "I BOKSEN,' SIGER HAN.":GOTO 1210
4520 ON NO-13 GOTO 4590,4560,4590,4530,4230
4530 IF G(9)<>5 THEN PRINT "'DEN ";UN$:GOTO 1210
4540 PRINT "'DET ER EN GANSKE ALMINDELIG":PRINT
      "REGNEMASKINE,' SIGER HAN."
4550 GOTO 1210

```

```

4560 PRINT "'DEN TILHØRER MIG. EAGLE KOM":PRINT "IND
      OG BAD MIG BRINGE DEN"
4570 PRINT "IND I BOKSEN, OG JEG BAR DEN":PRINT "IND
      TIL HAM OG RENFIELD,'"
4580 PRINT "SIGER HAN.":GOTO 1210
4590 PRINT "'DEN TILHØRER PRIAM,' SIGER HAN.":GOTO
      1210
4600 ON NO-13 GOTO 4660,4660,4660,4610,4230
4610 IF G(10)<>5 THEN PRINT "'DET ";UN#:GOTO 1210
4620 Y$="(FBHMF!EAEF!BG!FO!EPTJT"+V#+
      "EAEC SJOHFOEF!OJLPUJO"
4630 X$="PWFGASU!WJB!FO!QVTUFSASTQJM/"+V#+
      "IBO!MFWFEF!IAKTU!UJ!TFLVOEFS"
4640 Z$="FGUFS!BU!IBO!IBWEF!G^FU"+V#+
      "QJMF0!J!SZHHFO-(!TJHFS!IBO/"
4650 GOSUB 6110:X$="":Z$="":GOTO 1210
4660 PRINT "'DET ER MR. EAGLES LIG,' SIGER HAN.":
      GOTO 1210
4670 ON NO-13 GOTO 4790,4750,4790,4680,4230
4680 IF G(11)<>5 THEN PRINT "'DEM ";UN#:GOTO 1210
4690 IF G(19)=5 THEN 4720
4700 PRINT "'JEG KAN IKKE UDLEDE NOGET"
4710 PRINT "AF DEM UDEN SUPPLERENDE":PRINT
      "MATERIALE,' SIGER HAN.":GOTO 1210
4720 PRINT "'EN SAMMENLIGNING MED DEM OG":PRINT "DE
      PAPIRER, DER LA I BOKSEN,"
4730 Y$="WJTFS-!BU!QSJBN!IBS!CFH^FU"+V#+
      "VOEFSTMØC-(!TJHFS!IBO/":I4=1
4740 GOSUB 6110:GOTO 1210
4750 IF I4<>0 THEN 4770
4760 PRINT "'DET ER MIG, DER HAR FØRT DEM,'" :PRINT
      "SIGER HAN.":GOTO 1210
4770 Y$="(KFH!IBS!JLLF!CFH^FU!VOEFSTMØC/":GOSUB 6110
4780 PRINT "JEG NÆGTER ALT,' SIGER HAN.":GOTO 1210

```

```

4790 PRINT "'DET ER PRIAM, DER FØRER DEM,'" :PRINT
      "SIGER HAN.":GOTO 1210
4800 ON NO-13 GOTO 4840,4840,4840,4810,4230
4810 IF G(12)<>5 THEN PRINT "'DEM ";UN$:GOTO 1210
4820 PRINT "'DER ER INTET USÆDVANLIGT":PRINT "VED
      DEM,' SIGER HAN"
4830 PRINT "OG TILFØJER SMILENDE:":PRINT "'UD OVER
      INDHOLDDET.'" :GOTO 1210
4840 PRINT "'MR. EAGLE HAVDE DEM I EN REOL":PRINT
      "UDE I GANGEN. HAN HOLDT"
4850 PRINT "MEGET AF KRIMINALROMANER,'" :PRINT "SIGER
      HAN.":GOTO 1210
4860 ON NO-13 GOTO 4950,4890,4950,4870,4230
4870 IF G(13)<>5 THEN PRINT "'DEN ";UN$:GOTO 1210
4880 PRINT "'DET ER EN GANSKE ALMINDELIG":PRINT
      "NØGLE,' SIGER HAN.":GOTO 1210
4890 IF G(13)<>0 THEN 4940
4900 PRINT "'NØGLEN TIL BAGDØREN?":PRINT "DEN HAR
      JEG. JEG LASTE"
4910 PRINT "DØREN EFTER RENFIELD,'" :PRINT "DA HAN
      GIK. DØREN KAN IKKE"
4920 PRINT "LASES OP UDEFRA,' SIGER HAN":PRINT "OG
      GIVER DIG NØGLEN."
4930 G(13)=-1:GOTO 1210
4940 PRINT "'JEG HAR ALLEREDE":PRINT "GIVET DEM
      DEN,' SIGER HAN.":GOTO 1210
4950 PRINT "'NØGLEN TIL BAGDØREN?":PRINT "'DEN HAR
      PRIAM,' SIGER HAN.":GOTO 1210
4960 ON NO-13 GOTO 5250,4990,5110,4970,4230
4970 PRINT "'HVORFOR SPØRGER DE MIG OM HAM?"
4980 PRINT "JEG KENDER HAM IKKE,' SIGER HAN.":GOTO
      1210
4990 PRINT "'RENFIELD GIK SAMMEN MED EAGLE":PRINT
      "IND I BOKSEN. JEG BAR MIN"

```

```

5000 PRINT "REGNEMASKINE IND TIL EAGLE,":PRINT "OG
      RENFIELD OG JEG GIK UD AF"
5010 PRINT "BOKSEN. VI HØRTE BEGGE EAGLE":PRINT "SLA
      SLAEN FOR INDEFRA.":F1=1
5020 PRINT "SMITH KOM UD OG SAGDE FARVEL":PRINT "TIL
      RENFIELD, DER SKULLE HJEM,"
5030 PRINT "OG VILLE AFLEVERE NOGLE AKTIER"
5040 PRINT "TIL HUTCHINSON OG CO. PA VEJEN,"
5050 PRINT "OG JEG LUKKEDE DEREFTER RENFIELD"
5060 PRINT "UD GENNEM BAGDØREN OG LASTE DEN."
5070 PRINT "SMITH OG JEG GIK DEREFTER":PRINT "ATTER
      IND PA KONTORET, HVOR"
5080 PRINT "VI BEGGE BLEV, TIL VI BLEV":PRINT
      "BEKYMREDE FOR EAGLE. DA HAN IKKE"
5090 PRINT "ÅBNEDE DØREN TIL BOKSEN,":PRINT "DA VI
      BANKEDE PA, TILKALDTE VI"
5100 GOTO 5230
5110 PRINT "'RENFIELD GIK SAMMEN MED EAGLE":PRINT
      "IND I BOKSEN, OG ET ØJEBLIK"
5120 PRINT "EFTER KOM EAGLE IND TIL OS":PRINT "PA
      KONTORET OG BAD PRIAM BÅRE"
5130 PRINT "SIN REGNEMASKINE IND I BOKSEN."
5140 PRINT "SA HØRTE JEG RENFIELD OG PRIAM"
5150 PRINT "GA UD AF BOKSEN, OG JEG HØRTE":PRINT
      "TYDELIGT EAGLE SLA SLAEN FOR"
5160 PRINT "INDEFRA. JEG GIK UD OG SAGDE"
5170 PRINT "FARVEL TIL RENFIELD, DER SKULLE"
5180 PRINT "HJEM. JEG SA PRIAM LUKKE RENFIELD"
5190 PRINT "UD AD BAGDØREN OG LASE DEN.":F2=1
5200 PRINT "DEREFTER GIK PRIAM OG JEG":PRINT "ATTER
      IND PA KONTORET, HVOR"
5210 PRINT "VI BEGGE BLEV, TIL VI BLEV":PRINT
      "BEKYMREDE FOR EAGLE. DA HAN IKKE"
5220 PRINT "ÅBNEDE DØREN TIL BOKSEN,":PRINT "DA VI
      BANKEDE PA, TILKALDTE VI"

```

```

5230 PRINT "POLITIET, SOM SPRÆNGTE BOKSDØREN"
5240 PRINT "OG FANDT MR. EAGLES LIG":PRINT
      "DERINDE,' SIGER HAN.":GOTO 1210
5250 PRINT "'MR. EAGLE KALDTE MIG IND":PRINT "I
      BOKSEN OG BAD MIG GA OVER"
5260 PRINT "TIL HUTCHINSON OG CO. MED NOGLE"
5270 PRINT "AKTIER. JEG KOM AKTIERNE I MIN"
5280 PRINT "TASKE, OG DA PRIAM VAR KOMMET IND"
5290 PRINT "MED SIN REGNEMASKINE TIL EAGLE,"
5300 PRINT "GIK VI BEGGE UD SAMMEN, OG VI"
5310 PRINT "HØRTE HAM BEGGE SLÅ SLÅEN FOR.":PRINT
      "DERNÆST KOM SMITH OG SAGDE"
5320 PRINT "FARVEL TIL MIG, OG HAN OG":PRINT "PRIAM
      LUKKEDE MIG UD AD":F3=1
5330 PRINT "BAGDØREN, OG JEG AFLEVEREDE":PRINT
      "AKTIERNE TIL HUTCHINSON"
5340 PRINT "PA VEJEN HJEM,' SIGER HAN.":GOTO 1210
5350 ON NO-13 GOTO 5250,4990,5110,4960,4230
5360 GOTO 5350
5370 IF NO<>17 AND NO<>18 THEN 5400
5380 PRINT "'HOLD DEM VENLIGST TIL"
5390 PRINT "DE MISTÆNKTE PERSONER,':PRINT "SIGER
      HAN.":GOTO 1210
5400 PRINT "'KENDER HAM IKKE,' SIGER HAN.":GOTO 1210
5410 ON NO-13 GOTO 5490,5490,5490,5420,4230
5420 IF G(19)<>5 THEN PRINT "'DEM ";UN$:GOTO 1210
5430 IF G(11)=5 THEN 5460
5440 PRINT "'JEG KAN IKKE UDLEDE NOGET":PRINT "AF
      DEM UDEN SUPPLERENDE"
5450 PRINT "MATERIALE,' SIGER HAN.":GOTO 1210
5460 PRINT "'EN SAMMENLIGNING MED DEM OG":PRINT
      "KASSEBØGERNE FRA SKABET"
5470 Y$="WJTFS-!BU!QSJBN!IBS!CFH^FU"+V$+
      "VOEFSTMØC-(!TJHFS!IBO/*:GOSUB 6110
5480 I4=1:GOTO 1210

```

```

5490 PRINT "'DEM KENDER JEG INTET TIL,' SIGER
      HAN.":GOTO 1210
5500 ON NO-13 GOTO 5590,5590,5590,5510,4230
5510 IF FI=0 THEN PRINT "'FINGERAFTRYK? SADAN
      NOGEN":PRINT UN$:GOTO 1210
5520 IF PU<>0 THEN 5550
5530 PRINT "'JEG HAR IKKE MATERIALE NOK."
5540 PRINT "DE MA UD OG TAGE NØGLE":PRINT "FLERE,'
      SIGER HAN.":GOTO 1210
5550 IF SM<>1 THEN 5530
5560 Y$="(EFU!FS!TNJUIT!GJOHFSBGUSZL-"
5570 X$="EFS!TJEEFS!Q^!QVTUFSASFU-("+V$+"TJHFS!IBO/"
5580 GOSUB 6110:X$="":I1=1:GOTO 1210
5590 PRINT "'DEN SLAGS HAR JEG IKKE":PRINT "FORSTAND
      PA,' SIGER HAN.":GOTO 1210
5600 PRINT "'DE ER FORSVUNDET,' SIGER HAN."
5610 PRINT "'DET VED DE DA GODT, IKKE?":GOTO 1210
5620 GOTO 4440
5630 PRINT "'DET ER EN TRÆKKANAL,' SIGER HAN.":GOTO
      1210
5640 PRINT "'SPILLET?' SPØRGER HAN FORBAVSET."
5650 PRINT "'DET GAR UD PA AT FINDE MORDEREN."
5660 PRINT "DET TROEDE JEG DA, DE VIDSTE.":GOTO
      1210
5670 PRINT "'KAN DE IKKE BRUGE"
5680 PRINT "DERES ØJNE?' SPØRGER HAN."
5690 PRINT "'DE BEFINDER DEM "+S$+L$(LO)+"":GOTO
      1210
5700 ON NO-13 GOTO 5740,5740,5740,5710,5710
5710 PRINT "'DET ER DEM, DER ER DETEKTIVEN,'"
5720 PRINT "SIGER HAN. 'HAR DE VIRKELIG"
5730 PRINT "IKKE UNDERSØGT DET?":GOTO 1210
5740 PRINT "'VINDUET I GANGEN?' SIGER HAN."
5750 PRINT "'DET HAR IKKE VÆRET ABNET":PRINT "I
      UMINDELIGE TIDER.":GOTO 1210

```

```

5760 ON NO-13 GOTO 5820,5800,5820,5770,5770
5770 PRINT "'DET ER DEM, DER ER DETEKTIVEN,'"
5780 PRINT "SIGER HAN. 'HAR DE VIRKELIG"
5790 PRINT "IKKE UNDERSØGT DEN?'" :GOTO 1210
5800 PRINT "'DØREN I GANGEN?' SIGER HAN."
5810 PRINT "'DEN HAR JEG NØGLEN TIL.'" :GOTO 1210
5820 PRINT "'DØREN I GANGEN?' SIGER HAN."
5830 PRINT "'DEN HAR PRIAM NØGLEN TIL.'" :GOTO 1210
5840 PRINT "'DET ER ET VERDENSHJØRNE,'" :PRINT "SIGER
      HAN ";
5850 PRINT "OG KIGGER PÅ DIG," :PRINT "SOM OM DU ER
      ANDSSVAG." :GOTO 1210
5860 IF D$="EAG" THEN PRINT "'HAN ER DØD,' SIGER
      HAN." :GOTO 1210
5870 PRINT "'ET INTERESSANT SPØRGSMAL,' SIGER HAN."
5880 PRINT "'MEN HVAD PRÆCIS MENER DE"
5890 PRINT "MED ";CHR$(34)+D1$+CHR$(34)+"?"
5900 PRINT "DE BLIVER NØDT TIL AT":PRINT
      "OMFORMULERE SPØRGSMålet."
5910 PRINT "JEG BEKLAGER, MEN DET ER":PRINT "IKKE
      MIG, DER HAR AFSTUKKET"
5920 PRINT "REGLERNE FOR DETTE SPIL."
5930 PRINT "JEG DELTAGER BARE I DET!'" :GOTO 1210
5940 CLS:REM COMMODORE-BRUGER, SKRIV I STEDET: PRINT
      CHR$(147) *****
5950 PRINT "DU STAR "+L$(LO)
5960 IF AP=0 AND LO=2 THEN 5980
5970 GOTO 6000
5980 PRINT "PÅ BORDET FORAN DIG":PRINT "LIGGER DIN
      LOMMEBOG."
5990 PRINT "DEN VILLE DU GØRE":PRINT "KLOGT I AT
      UNDERSØGE."
6000 PRINT:PRINT "DU KAN GÅ:"
6010 FOR I=1 TO 4:IF L(LO,I)<>0 THEN PRINT V$(I)
6020 NEXT I

```



```

6030 RETURN
6040 PRINT
6050 PRINT "DU RANSAGER STEDET OG FINDER":PRINT "DE
      STJALNE DIAMANTER UNDER"
6060 PRINT "ET LØST GULVBRET. TIL LYKKE!"
6070 PRINT "DU HAR AFSLØRET MORDEREN!":PRINT
      "SPILLET ER FÆRDIGT.":END
6080 IF NO<14 OR NO>18 THEN PRINT "MEGET MORSOMT.":
      GOTO 1210
6090 IF G(NO)<>LO THEN PRINT "ER IKKE HER.":GOTO
      1210
6100 ON NO-13 GOTO 5250,4990,5110,5370,5370
6110 FOR I=1 TO LEN(Y$)
6120 PRINT CHR$(ASC(MID$(Y$,I,1))-1);
6130 NEXT I
6140 PRINT
6150 IF LEN(X$)<1 THEN RETURN
6160 FOR I=1 TO LEN(X$)
6170 PRINT CHR$(ASC(MID$(X$,I,1))-1);
6180 NEXT I
6190 PRINT
6200 IF LEN(Z$)<1 THEN RETURN
6210 FOR I=1 TO LEN(Z$)
6220 PRINT CHR$(ASC(MID$(Z$,I,1))-1);
6230 NEXT I
6240 PRINT
6250 RETURN
6260 PRINT "PRØV NOGET ANDET, ";NA$;"!":GOTO 1210
6270 PRINT "TRYK PÅ MELLEMRUMSTANGENTEN,"
6280 PRINT "NÅR DU VIL FORTSÆTTE.":RETURN

```


Stikordsregister

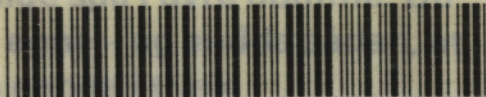
- Anagrammer 116
Bæreevne 110, 184
Datasæt 22, 28, 36
Dimensionering 24, 28
Disketter 231
Døre 26, 28
Egenskaber 109
Fortsætte-rutine 64, 90, 103, 111, 232
Gemme-rutine 64, 90, 103, 111, 184, 232
Genstande 24
Grundkommandoer 48
Grundtal 112
Gå-rutine 49
Handlingen 54
Hjælpe-rutine 73
Input 35, 42
Korttegning 120
Labyrinter 114, 186, 228
Linieinterval 34, 67
Linieskift 100, 232
Lokaliteter 22, 186
Lægge-rutine 18, 51
Navneord 35, 38, 43
Om igen-rutine 45
REM-linier 39
Real Time 104, 117
Retninger 26
Skærmbredde 99
Skærmpresentation 98
Skærmttekst-rutine 39
»Snyde«-labyrinter 114, 131, 180
Spilletid 104
Spilleur 108
Status-rutine 19, 20, 52
Store/små bogstaver 119
Strengdeling 43
Styrketal 110
Synonymer 74, 182
Tage-rutine 18, 50
Tidsfrist 104
Tilfældighedstals-
 generatoren 110, 186
»Tomme« løkker 88, 177
»Tomme« ord 70, 78
Tvangslægge-rutine 184
Udsagnsord 35, 38, 43
Æ 71, 118, 120, 179, 235
Ændrings-rutine 72, 103, 180
Ø 71, 118, 120, 235
Å 46, 71, 118, 120, 235

I denne bog lærer du at programmere
computereventyr i BASIC.

Har du en Commodore 64 eller en IBM PC, så
kan du skrive bogens programmer direkte af.
Har du en anden BASIC-dialekt, rummer bogen
så indgående forklaringer på programmerne, at
det ikke er noget problem at foretage en
tilpasning til netop din maskine.

Computer-eventyr er tekstspil, hvor computere
på skærmen forklarer, hvad der sker, og
spørger dig, hvad du vil gøre. Du befinder dig
måske i en ufremkommelig jungle eller et
skummelt slot, hvor du skal finde en skat og
samtidig forsvare dig mod alskens farer og
passe på, at du ikke farer vild.

Med denne bog ved hånden behøver du ikke
længere at købe færre titler, du kan
kan i stedet lave dine egne.
de aldrig bliver ens to.



258190327