

7-8/1989

KLUB MIKROELEKTRONIKY



ATARI®

Z P R A V O D A J

OLOMOUC

Vážení přátelé, uživatelé

ATARI ST!

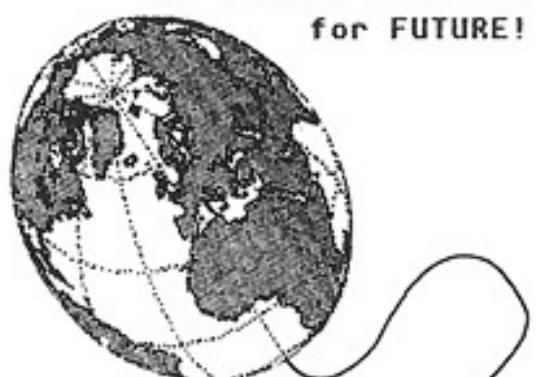
Chtěli bychom Vás informovat o zahájení činnosti skupiny uživatelů počítačů ATARI ST v rámci ATARI klubu Olomouc.

Dne 28. 10. 1989 se konala zahajovací schůzka zájemců, dne 4. 11. 1989 pak následovala první pravidelná schůzka. Zatím jsme se stihli vzájemně představit a začali jsme vytvářet databázi členů. Proběhla také diskuse o záměrech a cílech naší společné činnosti.

Zatím jsou mezi námi uživatelé z Olomouce, Přerova, Prostějova a dalších okolních měst i obcí. Naším prvním záměrem je vytvořit podrobnou databázi členů i programů, které mají členové k dispozici. Chceme se rovněž dovedět, jakým způsobem mohou členové přispět do společné činnosti, například znalostí jazyků, ovládáním některých rozšířených programů, odbornou literaturou, znalostí hardwaru, napojením na jiné kluby nebo organizace atd. Postupně bychom chtěli vytvořit archív odborné literatury i programů přístupný všem členům, využít publikační činnost a vytvořit základní hardwarové zázemí. Pro své členy chceme také pořádat osvětovou činnost - přednášky i praktický výcvik, které by byly zaměřeny na tvorbu i využití softwaru i na některé otázky hardwaru.

Případní zájemci se mohou přihlásit písemně na tuto adresu:

The "RIGHT" Computer
for FUTURE!

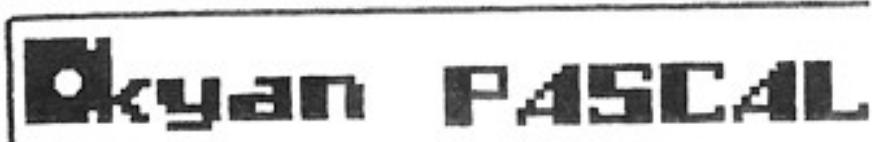


ATARI Klub
PS 137
OLOMOUC
772 11



V přihlášce uveďte tyto údaje:

- jméno a příjmení
 - adresu s PSČ
 - rodné číslo
 - povolání
 - zaměstnavatele
 - telefon (doma i do zaměstnání)
 - členství ve Svazarmu (ano (uveďte číslo průkazu) - ne)
 - čím můžete přispět k naší společné činnosti
 - jakou počítačovou konfiguraci vlastníte
(typ počítače, monitoru, disketové jednotky, tiskárny, ...)
- Pravý horní roh obálky označte písmeny ST.



Kyan Software, Inc.

1850 Union Street #163
San Francisco, CA 94123
U.S.A.

Od Kyan PASCALu k Turbo PASCALu!

ins. Jaroslav Choma

Cást 6.

Už po šesté se setkáváme s jazykem Kyan PASCAL. Doufám, že pro mnohé z vás právě tyto články byly impulsem pro první pokusy. V předcházející části jsme si ukázali řešení některých procedur určených pro práci se soubory. Měli bychom si ještě objasnit jejich používání. První uvedená procedura Delete je určena pro vymazání zadaného souboru z diskety nebo RAMdisku. Dále jsou uvedené procedury Rename na přejmenování souboru, Protect pro zablokování souboru před smazáním, UnProtect pro zrušení zablokování. Uvedené jsou i procedury pro formátování - SFormat pro formátování jednoduchou hustotou a DFormat na formátování dvojnásobnou (medium) hustotou. Zde poznamenávám, že v případě, kdy budeme využívat současně více procedur, je výhodnější je mít v jednom boxu, ale délku pole pro název je nutné určovat podle nejdélšího použitého souboru, případně podle změn pro Rename. Při přenosu těchto procedur pro verzi 2.0 bude pravděpodobně výhodnější nadefinovat více typů - zvlášť pro Rename a zvlášť pro ostatní procedury. Pozor, změnu potom musíme udělat i v záhlaví procedur!

Přenositelnost programů na jiné počítače

U našich procedur jsme zapomněli na jeden mimořádně dôležitý problém, a to problém přenositelnosti programů do jiných verzí jazyka PASCAL, např. pro Turbo PASCAL. Zde se musíme na chvíli zastavit, protože pochopení této části bude klíčové pro ty, kteří chtějí programovat v jazyku Kyan PASCAL a těchto programů později využít pro jiné typy počítačů s jiným interpretorem jazyka PASCAL (IBM PC, SMEP....). Co si nutně musí uvědomit každý programátor, který si stanoví tento cíl?

Musíme si totiž uvědomit, že v současnosti existuje mnoho verzí jazyka PASCAL pro různé typy počítačů, dokonce pro stejný typ bylo vyvinuto několik odlišných verzí jazyka PASCAL (viz

IBM PC). Bez obav tedy mohu prohlásit, že vytvořit bezproblémový přenositelný program je mimořádně náročné a někdy prakticky nemožné. Ve kterých oblastech tedy můžeme čekat zásadní problémy?

- 1) Jsou to odchylky od normy ISO, která popisuje standartní jazyk PASCAL. Samozřejmě musíme znát odchylky i u verze Kyan PASCAL. Měli bychom však poznat nejen omezení, ale i rozšíření možností jazyka na jistém typu počítače, které může být při vývoji našeho programátorského umění velmi nebezpečné. Rozšíření funkcí jazyka nás láká tím, že si šetříme práci s vypisováním a vymýšlením standartních procedur.
- 2) Je nutné dokonale poznat rozdíly v implementacích verzí, pro kterou vyvíjíme procedury a funkce, a verze, se kterou pracujeme. Typickým příkladem je právě Turbo PASCAL, který nezná procedury put, set. Ty jsou nahrazeny standartními procedurami read a write. Velké problémy mohou vzniknout při zpracovávání řetězců. Zde naopak jazyk Kyan PASCAL představuje standartní způsob práce a jazyk Turbo PASCAL se dosti liší od standartu, protože dovoluje používání předdefinovaného typu dat strings.
- 3) Je třeba pochopit rozdíly v operačních systémech, protože ty mimořádně ovlivňují efektivnost algoritmů – to, co jednomu překladači vyhovuje, na druhém může působit nesmírné problémy. Specifické situace může působit používání různých "volných" registrů v operačním systému. Například různé verze DOSu pro ATARI využívají těchto registrů pro své (tedy odlišné) účely...
- 4) Neměli bychom využívat některé specifické vlastnosti nebo chyby překladače. Právě námi uvedená procedura Delete() je tímto trestuhodným příkladem. Zdá se, že právě tento problém se dost obchází (vzniká skupina programátorů typu "tajemství si odnesu až do hrobu"). Ospravedlněním pro to není ani urychlení ani zkrácení programu...
- 5) Použití assembleru je u těchto programů nejtěžším ohříškem. Assembler totiž může při neodborném použití obcházet zábrany, které jsou pro PASCAL typické a jsou jeho největší výhodou.
- 6) Odchylky, které nejsou popsány v normě jazyka, ale pro programátora jsou podstatné, např. přesnost výpočtů, rozsah číselných hodnot proměnných atd. Mezi příklady, které byly uvedené v našem kursu, tyto vlastnosti využívaly např. funkce Int(), Frac() a procedura Truncate().

O těchto problémech bychom mohli popsat mnoho stránek, ale život mnohokrát přináší situace, které jsou ve srovnání s popsánými jevy daleko větším horrorem... Vráťme se proto k našemu tématu – jazyku Kyan PASCAL.

Podstatné rozdíly verzi 1.X a 2.X

Úvodem této kapitoly bych rád poděkoval kolesovi z AK Olomouc, panu Hrdličkovi, který mi laskavě zapůjčil manuál pro verzi 2.0. Bez jeho pochopení by tato část nemohla vzniknout.

Jak můžeme hodnotit novou verzi 2.0? Stručně si řekněme o nejdůležitějších rozdílech:

1. Odstraněné chyby známé z verze 1.X (chyba při aritmetických operacích, špatná kontrola shodnosti typu, kontrola počtu parametrů).
2. Nový knižní modul LIB, který je rozsáhlejší a nachází se od adresy \$8C00 po \$BBFF. Z toho vyplývá menší rozsah volné paměti. Rozsah se zmenšil na cca. 27 KB. Tento nedostatek je ale vyvážený podstatně hutnějším kódem po komplikaci.
3. Celý systém PASCALu je řízený z menu KIX (přitom není nutné opustit PASCAL).
4. Maximální délka identifikátoru se zvětšila na 256 znaků.
5. V identifikátorech je možné použít symbol podtrhnutí. Ten slouží k oddělení slov ve víceslovném identifikátoru.
6. Ve verzi 2.0 není implementována procedura assign. Je nahrazená funkcí pointer. Použití si můžeme ukázat na příkladě čtení z paměťového místa:

```
program cti;
var uk: char;
begin
  uk:=pointer(84);
  write(uk )
end.
```

Podobně můžeme vytvořit program pro zápis do paměťového místa

```
program zapis;
var uk: char;
begin
  uk:=pointer(84);
  uk := a
end.
```

7. Zavedená je nová funkce address(), jejímž výsledkem je umístění prvního bytu proměnné v paměti.
8. Implementována je i procedura page, která slouží k přechodu na novou stránku.
9. Změněná je syntaxe assembleru a také jsou podstatně rozšířeny jeho možnosti. Můžeme také používat malá písmena.
10. Podstatně je zdokonalena analýza chyb při překladu (140 indikovaných chyb pro PASCAL proti 35 chybovým hlášením u verze 1.X a 17 indikovaných chyb pro assembler proti 7 chybovým hlášením u verze 1.X).
11. Podstatnou změnou je i rozdělení komplikace na dvě etapy. V první etapě se provádí překlad do macroassembleru a v druhé etapě do vykonávatelného kódu. Přechod z fáze překladu

macroassembleru do PASCALu je automatický. Volbami je možné modifikovat parametry pro komplikaci. Výsledky po překladu do macroassembleru a do PASCALu jsou ukládány do dvou předdefinovaných souborů a.out a p.out.

12. Máme možnost využít ladící prostředky na úrovni strojového kódu.

13. Podstatný rozdíl je ve způsobu práce s proměnnými v assembleru. Proměnné nejsou umístěny 3 byty nad SP, ale až 5 bytů!.

14. Výrazný rozdíl je i při použití grafiky. Na rozdíl od verze 1.X je nutno grafiku nadefinovat na začátku programu:

```
#a
origin equ $4000
#
```

Změnou origin můžeme posouvat začátek programu v paměti a tak regulovat jeho umístění nebo oddělit chráněný prostor atd.

Velkou výhodou, kterou poskytují autoři této verze Kyan PASCALu, je dodání macro instrukcí, které realizují částečné algoritmy na disketě ve zdrojovém tvaru. Takto může každý zájemce využívat po nastudování jednotlivé macra ve svých programech.

Popis KIX - systému ovládání Kyan PASCALu 2.0

Změnou, která potěší každého zájemce o práci v PASCALu, je systém jeho řízení. U verze 1.X jsme měli možnost vyvolat DOS a po vykonání určité činnosti jsme se vraceли zpět do PASCALu pomocí volby B. Pro verzi 2.X byl vytvořen systém nazvaný KIX. Ten obsahuje tyto moduly:

CD - umožňuje změnu nastavení aktuální disketové mechaniky. Povel je v tvaru:

CD D8: (změna aktuální mechaniky na D8:)

LS - umožňuje výpis z diskety na dané zařízení. Povel má tvar

LS D1: >P: (vylistuje obsah z D1: a vypíše ho na tiskárně)

CAT - zobrazí obsah souboru na daném zařízení. Povel má tvar:

CAT D1:meno >P: (vypíše obsah souboru na tiskárně)

U všech povelů tohoto typu platí, že jestliže nejdáme typ zařízení, automaticky se nadefinuje obrazovka E:

CP - umožňuje kopírování souborů. Povel má tvar:

CP vstupní výstupní soubor (za vstupní dosadíme jméno souboru, který chceme kopírovat, a za výstupní dosadíme jméno, které má mít soubor po zkopirování)

MV - slouží k přejmenování souboru. Povel má tvar:

MV staré nové jméno (staré představuje jméno programu, které chceme přejmenovat, a nové představuje nové jméno souboru)

RM - umožňuje vymazávání zvoleného souboru. Povel má tvar:

RM jméno (soubor daného jména bude po zadání vymazán)

CHMOD - povel je určen pro změnu ochrany souboru před smazáním.

Tvar povelu:

CHMOD -W jméno (je to ekvivalent lock)

CHMOD +W jméno (je to ekvivalent unlock)

FORMAT - povel slouží k formátování diskety na zvolenou hustotu. Povel má tento tvar:

FORMAT D1: D (formátuje disketu v mechanice 1 na rozšířenou hustotu)

FORMAT D1: S (liší se pouze volbou jednoduché hustoty)

Co se týká chyb - znova je nedokonale vyřešeno nastartování programu po zadání! Program se nespustí, ale znova sa natáhne do paměti. Celkově je však tato verze PASCALu velmi zdařilá. Její podstatnou výhodou je její přístupnost, a proto časem určitě vznikne dost zajímavých a užitečných programů pro tuto verzi. Kyan PASCAL má tedy šanci stát se využívaným jazykem. Zřejmě se bude využívat pro náročnější programování (řešení problémů se složitou logikou) a pro vyučování, protože 8-bitové počítače ATARI se staly velmi populárními i na školách.

Cást č.7

Máme za sebou šest částí povídání o možnostech jazyka Kyan PASCAL. Mnoha zájemcům, především těm, kteří již měli s jazykem Kyan PASCAL určité zkušenosti, snad tento seriál ukázal nové využití tohoto jazyka. Určitě však existuje početná obec Ataristů, kteří ještě váhají a nevědí, jak začít...

Tém váhajícím jsem na závěr mého souboru článků připsal tuto kapitolu. Začneme od začátku - tak, jak se převážná část uživatelů seznámila s programováním na počítači Atari XL/XE - s jazykem BASIC. Skalní zástupci jazyků PASCAL a BASIC teď jistě kroutí hlavou, ale protože principy velké části programovacích jazyků jsou shodné, můžeme si takové spojení dovolit.

Jazyk BASIC vznikl v roce 1964 a od té doby prodělal obrovský vývoj. Rozšířil se z několika důvodů - jednak vyžaduje pro práci malou paměť a jednak poskytuje programátorům-amatérům zrádně lehké interaktivní ladění programů. Po vzniku jazyka PASCAL v roce 1971 se mnohé myšlenky ze struktury PASCALu začaly objevovat v různých implementacích BASICu. Skutečný rozmach zaznamenal BASIC v 80. letech na osobních počítačích. Zde se ale začaly naplna projevovat záporny tohoto jazyka. Firmy to vyřešily po svém - začaly "lepit" k jazyku nové příkazy a funkce s takovým úspěchem, že by ani tvůrcové BASICu tyto výtvory nepoznali. Odstraňujícím příkladem těchto verzí jsou

verze HP, které zabírají i více než 200 KB! Podle vzoru PASCALu se mnoho verzí tohoto jazyka stalo plně strukturovanými a z BASICu zůstal vlastně jen název...

Podobně jako mnoho jiných značek počítačů i počítače ATARI disponují těmito strukturovanými jazyky, např. Turbo BASIC XL, BASIC XL, BASIC XE a další. Nejrozšířenějším se stal vzhledem ke své dostupnosti vynikající Turbo BASIC XL. Tato verze má všechny vlastnosti, které potřebujeme pro vysvětlení některých struktur, a proto môžeme začít.

Učme se kyan PASCAL s Turbo BASICem!

Jazyk BASIC vznikl jako nejjednodušší programovací jazyk. Vycházel z principů, které byly známy v době jeho vzniku. Tvůrce jazyka Turbo BASIC XL, Frank Ostrowski, ponechal všechny vlastnosti standardního BASICu (vzhledem ke služitelnosti) a přidal nové výmoženosti. Touto částí článku bych rád vyšel v ústřední programátorem, kteří aktivně znají Turbo BASIC XL a nenásilnou formou by chtěli začít s jazykem PASCAL. Před samotným úvodem do této problematiky si musíme však stanovit několik zásad, které nám ulehčí programování:

- 1) Měli bychom zapomenout na existenci příkazu GOTO! (Respektování tohoto zákona nám bude ze začátku dělat problémy, ale musíme si uvědomit, že Turbo BASIC poskytuje takové možnosti, které vylučují nutnost použití skoku.)
- 2) V případě, kdy neumíme vytvořit konstrukci programu bez skoku, je dobré použít návěsti a skok realizovat přes toto návěsti (tuto možnost však doporučujeme využívat co nejméně).
- 3) Místo příkazu GOSUB používat volání podprogramů pomocí EXEC.
- 4) Používat konstrukci IF...ENDIF, IF...ELSE...ENDIF a IF...THEN. Ale ne pro skok na řádek!
- 5) Nepoužívat smyčku FOR...TO...NEXT ve variantě s podmínkou, pomocí které chceme cyklus opustit při splnění podmínky. V těchto případech bychom měli použít REPEAT...UNTIL (podmínka), anebo WHILE (podmínka)...WEND.
- 6) Pro označování proměnných bychom měli využívat i viceslovné názvy a jednotlivá slova oddělovat symbolem podtržení. Viceslovné názvy zvyšují přehlednost textu programu a lépe vystihují obsah proměnné.
- 7) Na jeden řádek nepsat více než jeden příkaz (pro lepší přehlednost a orientaci v programu).

Konstrukce Kyan PASCALu a Turbo BASICu XL

V této části si uvedeme ekvivalentní konstrukce těchto jazyků tak, aby mohly sloužit při výuce jazyka PASCAL. Vždy v názvu uvedeme konstrukci z Kyan PASCALu a potom její ekvivalent v Turbo BASICu XL.

if...

Konstrukce může mít více variant:

1. if...then...;

V Turbo BASICu bude mít konstrukce tvar IF...THEN..., kde za THEN bude jeden přířaďovací příkaz.

2. if...then

begin

.

.

end;

V Turbo BASICu bude mít tvar IF...ENDIF, kde podobně jako v PASCALu mezi begin a end i mezi IF a ENDIF může být více příkazů.

3. if...then...else...;

V Turbo BASICu dostaneme ekvivalent IF...ELSE...ENDIF. V případě použití ELSE v Turbo BASICu je použití ENDIF povinné! Modifikaci konstrukce v PASCALu jsou varianty, kde mezi then a else je více příkazů, stejně tak po else, např:

if...then

begin

.

.

end

else...anebo

if...then...

else

begin

.

.

end;

Použití takového rozhodovacího bloku v Turbo BASICu fakticky umožní odstranit skoky.

repeat...until...

Cykly s podmínkou patří mezi nejvýznamnější konstrukce jazyka PASCAL. Při jejich použití musíme dát pozor na

správnou definici podmínky, při které se má cyklus ukončit. Při zadání špatné definice totiž vzniká nekonečný cyklus. Ekvivalentem v jazyku Turbo BASIC je konstrukce REPEAT...UNTIL..., kde mezi REPEAT a UNTIL může být jeden příkaz nebo více příkazů. Přípustná je i varianta bez příkazu. Například ekvivalent konstrukce: repeat until keypressed; , kde jsme vlastně nadefinovali čekání na stlačení klávesy.

Při použití této konstrukce si musíme zapamatovat, že cyklus bude vykonaný minimálně jednou, protože test splnění je na konci cyklu. Existuje však i možnost testování na začátku cyklu. V případě, že cyklus se nesmí vykonat, pokud není splněna vstupní podmínka, je vhodná tato konstrukce:

```
while...do...
-----
```

Jazyk PASCAL připouští za "do" jeden příkaz, více příkazů nebo prázdný příkaz. V Turbo BASICu má ekvivalent v konstrukci WHILE...WEND.

Abyste viděli možnost strukturovaného programování v Turbo BASICu, ukážeme si řešení problému (je běžný v praxi a dá se využít i studující mládeži) – regresivní analýzu dat.

Častým problémem při výhodnocování měření nebo při statistických rozborech je na základě zjištěných, naměřených údajů vytvořit závislost ve tvaru vzorce. Nejčastěji se pro tyto účely používá metoda nejmenších čtverců. Metoda je popsána ve všech středoškolských učebnicích, a proto se ji nebudeme podrobně zabývat. Ukážeme si řešení programu v Turbo BASICu, který realizuje daný problém:

```
10 REM Program pro regresivní závislosti
20 CLS
30 REPEAT
40   ERR=0
50   TRAP 70
60   INPUT "Počet bodů=";N
70 UNTIL N>=3 OR ERR=0
80 DIM X(N),Y(N),R(N),Q(N),P$(1)
90 TRAP 40000
100 CLS
110 FOR I=1 TO N
120   PRINT "X(";I;")=";
130   INPUT POM
140   X(I)=POM
150   PRINT "Y(";I;")=";
160   INPUT POM
170   Y(I)=POM
180 NEXT I
190 REPEAT
200   REPEAT
210     CLS
220     ERR=0
230     R=0
```

```

240      TRAP 320
250      PRINT "Zvol funkci:"
260      PRINT "1-Linearni regrese"
270      PRINT "2-Exponencialni regrese"
280      PRINT "3-Logaritmicka regrese"
290      PRINT "4-Mocninna regrese"
300      PRINT "5-Ukonceni programu"
310      INPUT "Funkce ";R
320      UNTIL ERR>0
330      IF R<5 AND R>=1
340          A=0
350          B=0
360          C=0
370          D=0
380          E=0
390          F=0
400      FOR I=1 TO N
410          R(I)=X(I)
420          IF (R=3 OR R=4)
430              POM=X(I)
440              R(I)=LOG(POM)
450          ENDIF
460          Q(I)=Y(I)
470          IF (R=2 OR R=4)
480              POM=Y(I)
490              Q(I)=LOG(POM)
500          ENDIF
510          A=A+R(I)
520          B=B+Q(I)
530          C=C+R(I)*R(I)
540          D=D+R(I)*Q(I)
550      NEXT I
560      G=(D-A*B/N)/(C-A*A/N)
570      H=(B-G*A)/N
580      FOR I=1 TO N
590          E=E+(R(I)-A/N)^2
600          F=F+(Q(I)-B/N)^2
610      NEXT I
620      CLS
630      IF R=1
640          PRINT "Rovnice y=a+b*x"
650          PRINT "a=";H;" b=";G
660      ENDIF
670      IF R=2
680          H=EXP(H)
690          PRINT "Rovnice y=a*exp(b*x)"
700          PRINT "a=";H;" b=";G
710      ENDIF
720      IF R=3
730          PRINT "Rovnice y=a+b*log(x)"
740          PRINT "a=";H;" b=";G
750      ENDIF
760      IF R=4
770          H=EXP(H)

```

```

780      PRINT "Rovnice y=ax+b"
790      PRINT "a=";H;" b=";G
800      ENDIF
820      K=G*SQR(E/F)
830      PRINT "Korelační koeficient ";K
840      PRINT "Pokud jsi precal, stiskni <RETURN>";
850      INPUT P$
870      ENDIF
880 UNTIL R=5
890 TRAP 40000
900 END

```

Uvedený program je úplně typickým řešením problému pro jazyk PASCAL, ale my jsme ho realizovali v Turbo BASICu. Ani skalní zastánce PASCALu by neměl mít námítky, protože v programu jsou dodrženy všechny pravidla pro PASCAL. Aspoň stručně bychom se měli zmínit o některých částech programu.

Rádky 30 až 70 slouží na zadání počtu dvojic naměřených údajů. Využijeme chybové ošetření pro zadání nečíselných údajů a testovaný je i rozsah menší než 3 dané dvojice. To je totiž podmínka pro správný chod programu. Horní rozsah není testován, ale to není žádný problém. Musíme si uvědomit, že zde uvedené programy jsou pouze polotovary určené k dalšímu zpracování.

Rádky 110 až 180 zabezpečují snímání naměřených údajů a jejich umístění v polích.

Rádky 190 až 900 tvoří hlavní část programu, kde se řeší celá logika. Můžeme volit jednu ze 4 regresivních závislostí a potom výpočet opakovat pro stejné údaje. Na výstupu dostaneme koeficienty regresivních závislostí a korelační koeficient, který v podstatě udává stupeň splnění závislosti od ideální závislosti. Může dosáhnout největší hodnotu 1. Uvedený program se dá velmi lehce přepsat do PASCALu. Z použitých prostředků bude samozřejmě nutné vyloučit dynamické deklarace, chybová ošetření a také bude třeba vytvořit mocninnou funkci. Ostatní rozdíly jsou zanedbatelné. Ještě je potřebné zdůraznit, že PASCAL umožňuje řešit některé části efektivněji s využitím funkcí a procedur. V tomto programu také nevidíme výhodu lokálních a globálních parametrů.

Závěr

Měli bychom udělat tečku za našimi setkáními s jazykem Kyan PASCAL. Je těžké zvolit vhodnou formu, vždyť jsme se střetávali dlouhou dobu. Na závěr chci vyzvat všechny skutečné zájemce o programování – programujte, uveřejňujte a šířte osvětu tak, aby se nám počítače staly skutečnými pomocníky. Obrazně řečeno: vlak nám ujel, ale proto snad nebudeme naříkat. Naopak se ho budeme snažit dosáhnout, a to všemi dostupnými prostředky. Proto ať žije ATARI a Kyan PASCAL!

Třídění tabulky čísel

Andrzej Biazik

S potřebou třídění čísel podle velikosti se setkal nebo setká každý uživatel počítače. Pro řešení tohoto úkolu existuje řada algoritmů. Od těch jednoduchých až po složité, které jsou ale mnohem rychlejší. Někdy vystačíme s jednoduchým algoritmem, ale většinou nás zajímá co nejrychlejší vytřídění čísel.

Protože rychlosť činnosti programu ovlivňuje nejenom algoritmus, ale i programovací jazyk, který programátor používá, nejlepších výsledků dosáhneme použitím vhodné procedury v asembleru. Pro ty, kteří nemají mnoho zkušeností v programování procesoru 6502, uvádíme proceduru ve strojovém kódu, která realizuje algoritmus třídění pomocí vkládání. Je to poměrně jednoduchá, ale přitom dostatečně rychlá procedura. Vlastní algoritmus můžete nalézt téměř v každé publikaci, která se zabývá programováním.

Výhody použití asembleru (pokud jde o rychlosť) ukáže nejlépe srovnání rychlosti vytřídění tabulky s 200 prvků při použití jazyků Atari BASIC a TURBO BASIC XL:

Atari BASIC	322,2 s
TURBO BASIC XL	134,2 s
Asembler	4,5 s

Rozdíl je evidentní.

Po tomto poněkud delším úvodu je čas věnovat se praktickému využití programu. Protože program zabírá více než jednu stránku paměti, byl umístěn do paměti od adresy 39424 (\$9A00), tedy před začátek obrazové paměti (mod GRAPHICS 0). To znamená, že použití grafiky s vyšší rozlišovací schopností nutně zničí proceduru. Ta není přemístitelná vzhledem k použití několika vnitřních odvolání, i když po určitých úpravách by bylo možno umístit program jinde.

Program je napsán v jazyku Atari BASIC s rutinou v asembleru. Program natypujte a zkontrolujte jeho správnost pomocí programu TYPO II.

Vývolání procedury třídění prováděme z BASICu pomocí příkazu USR. Syntaxe: X=USR(39424,TYP), kde proměnná TYP nabývá tyto hodnoty:

TYP=0 ... pro vzestupné třídění

TYP=1 ... pro sestupné třídění

Pozor, jiné hodnoty proměnné TYP mohou vést ke zhroucení programu.

Tříděna je první nalezená jednorozměrná tabulka a jako zpětný parametr je předáván počet tříděných prvků (nebo 0, jestliže tabulka není nalezena). Pro maximální zkrácení času je během třídění vypínán DMA pro ANTIC - "zhasnutí" obrazovky je tedy normálním jevem při práci procedury. Upozorňuji, že třídění se nezúčastní prvek tabulky o indexu 0.

ing. Jeroným Liška
Jiří Hrdlička

Listing 2 - assembly code:

```

0100      * = 39424
0110 FR0      = 212
0120 FR1      = 224
0130 DMA      = 559
0140 J       = 203
0150 I       = 205
0160 N       = 207
0170 VVTP     = 134
0180 STMTAB   = 136
0190 STARP    = 140
0200 POCZ    = 112
0210 BUF     = 114
0220 ADJ     = 36
0230 ADI     = 38
0240 IAD    = 40
0250 ;
0260 ; Třídění tabulky
0270 ; metodou vkládání
0280 ;
0290 ; (c) 1988 Andrzej Biazik
0300 ;
0310 PLA
0320 PLA
0330 PLA      ;zjištění
0340 ASL A      ;požadovaného
0350 ASL A      ;sledu
0360 ASL A      ;třídění
0370 ASL A
0380 ASL A
0390 ADC #16
0400 STA ZAP
0410 LDA VVTP    ;přepis počátek
0420 STA J      ;tabulky hodnot
0430 LDA VVTP+1  ;proměnných
0440 STA J+1    ;(VVTP)
0450 SZU LDY #0
0460 LDA (J),Y  ;hledej
0470 CMP #65    ;deklarovanou
0480 BEQ SPR    ;tabulkou
0490 DOD CLC
0500 LDA J      ;když tabulkou
0510 ADC #8    ;nenajdeš,
0520 STA J      ;ověř
0530 LDA J+1    ;následující
0540 ADC #0    ;proměnnou
0550 STA J+1
0560 SEC
0570 LDA J+1
0580 CMP STMTAB+1
0590 BCC SZU    ;zjisti,
0600 LDA J      ;zda nebyl

```

0610	CMP STMTAB	; překročen
0620	BCC SZU	; rozsah VVTP
0630	LDA #0	
0640	STA FR0	; jestliže ano,
0650	STA FR0+1	; potom konec programu
0660	RTS	; s vypsáním hodnoty 0
0670	SPR LDY #6	
0680	LDA (J),Y	; zjistit,
0690	CMP #1	; zda má tabulka
0700	BNE DOD	; jeden rozměr
0710	LDA #0	
0720	STA DMA	; vypni DMA
0730	LDY #2	
0740	CLC	
0750	LDA (J),Y	; vypočítej
0760	ADC STARP	; skutečnou
0770	STA POCZ	; adresu počátku
0780	INY	; tabulkys
0790	LDA (J),Y	
0800	ADC STARP+1	
0810	STA POCZ+1	
0820	INY	; zapíš do buňky
0830	LDA (J),Y	; N a N+1
0840	STA N	; počet
0850	INY	; tříděných
0860	LDA (J),Y	; prvků +1
0870	STA N+1	
0880	CLC	
0890	LDA POCZ	; urči
0900	ADC #12	; počáteční
0910	STA ADJ	; parametry
0920	LDA POCZ+1	; před
0930	ADC #0	; vstupem
0940	STA ADJ+1	; do smyčky
0950	LDA #2	
0960	STA J	
0970	LDA #0	
0980	STA J+1	
0990	POW SEC	; počátek
1000	LDA J	; hlavní smyčky
1010	SBC #1	; J se mění
1020	STA I	; od 2 do N
1030	LDA J+1	
1040	SBC #0	
1050	STA I+1	
1060	SEC	
1070	LDA ADJ	; skorekce
1080	STA IAD	; adres
1090	SBC #6	; právě
1100	STA ADI	; ověřovaných
1110	LDA ADJ+1	; prvků
1120	STA IAD+1	; tabulkys
1130	SBC #0	
1140	STA ADI+1	

```

1150 LDY #5
1160 ET1 LDA (ADJ),Y ;dočasné
1170 STA BUF,Y ;uchování
1180 DEY
1190 BPL ET1
1200 NRS LDX #5 ;vnitřní smyčka
1210 ET2 LDA BUF,X ;I se mění
1220 STA FR0,X ;od J-1 do 1
1230 DEX ;s krokem -1
1240 BPL ET2
1250 LDY #5 ;přepis
1260 ET3 LDA (ADI),Y ;prvky
1270 STA FR1,Y ;tabulky
1280 DEY ;do registrů
1290 BPL ET3 ;FR0 a FR1
1300 JSR ODE
1310 ZAP BPL EXT
1320 LDY #5 ;je-li výsledek
1330 ET4 LDA (ADI),Y ;porovnání kladná,
1340 STA (IAD),Y ;potom
1350 DEY ;opustí smyčku.
1360 BPL ET4
1370 SEC
1380 LDA ADI ;korekce
1390 STA IAD ;platných
1400 SBC #6 ;ukazatelů
1410 STA ADI
1420 LDA ADI+1
1430 STA IAD+1
1440 SBC #0
1450 STA ADI+1
1460 DEC I ;zmenší
1470 BNE NAS ;čítač;
1480 DEC I+1 ;není-li
1490 BPL NAS ;roven 0,
1500 CLC ;opakuj
1510 LDA POCZ
1520 ADC #6
1530 STA IAD
1540 LDA POCZ+1
1550 ADC #0
1560 STA IAD+1
1570 EXT LDY #5 ;konec
1580 ET5 LDA BUF,Y ;vnitřní smyčky
1590 STA (IAD),Y
1600 DEY
1610 BPL ET5
1620 CLC
1630 LDA ADJ
1640 ADC #6
1650 STA ADJ
1660 LDA ADJ+1
1670 ADC #0
1680 STA ADJ+1

```

1690	INC J		2230	BNE H02
1700	BNE PRW		2240	INX
1710	INC J+1		2250	CPX #6
1720	PRW SEC	; ověř, zda	2260	BNE ET7
1730	LDA J+1	; J<=N	2270	H01 LDA FR0
1740	CMP N+1		2280	BMI ET9
1750	BCS CZY	; jestliže ano,	2290	ET8 LDA #0
1760	JMP POW	; pak skoč	2300	CLD
1770	CZY SEC	; na začátek	2310	RTS
1780	LDA J	; smyčky	2320	H02 BCS H01
1790	CMP N		2330	LDA FR0
1800	BCS KON		2340	BMI ET8
1810	JMP POW		2350	ET9 LDA #128
1820	KON SEC		2360	CLD
1830	LDA N	; jestliže ne,	2370	RTS
1840	SBC #1		2380	.END
1850	STA FR0	; pak přepiš		
1860	LDA N+1	; N do FR0		
1870	SBC #0			
1880	STA FR0+1			
1890	LDA #34			
1900	STA DMA	; zapni DMA		
1910	RTS			
1920	ODE SEC	; začátek		
1930	LDA FR0	; porovnávací		
1940	CMP FR1	; procedury		
1950	BEQ ET6			
1960	PHP			
1970	PLA			
1980	AND #1			
1990	PHR			
2000	LDA FR0	; zjišťování		
2010	BPL HP1	; velikosti		
2020	LDA FR1	; a		
2030	BPL HP2	; znaků		
2040	PLA	; porovnávaných		
2050	HP4 ROR A	; čísel		
2060	ROR A			
2070	RTS			
2080	HP2 PLA			
2090	LDA #128			
2100	RTS			
2110	HP1 LDA FR1	; výstup		
2120	BPL HP3	; z procedury		
=130	PLA	; s nastaveným N		
2140	RTS	; když číslo		
2150	HP3 PLA	; z FR0<FR1		
2160	EOR #1			
2170	JMP HP4			
2180	ET6 LDX #1			
2190	SED			
2200	ET7 SEC	; porovnávání		
2210	LDA FR0,X	; následujících		
2220	CMP FR1,X	; čísel		

Listing 1 - program MLSORT.BAS

Part : 1

```

ZG 1 REM
ON 2 REM [ ML SORT ]
AL 3 REM
XI 4 REM by Andrzej Biazik
BO 5 REM (c) 1988 BAJTEK
UR 6 REM Call machine routine:
YJ 7 REM X=USR(39424,TYPE)
RG 8 REM TYPE=0 ASCENDING
DT 9 REM TYPE=1 DESCENDING
NF 10 A=39424:FOR I=0 TO 37
RJ 20 S=0:FOR J=0 TO 9:READ B:POKE A+J,B:S=S+B:NEXT J
VC 30 READ SK:IF S>SK THEN 60
RE 40 A=A+10:NEXT I
VG 50 ? "Ready.":END
PS 60 ? "Error in line ";100+10*I
IY 70 LIST 100+10*I:END
ZG 100 DATA 104,104,104,10,10,10,10,10,105,16,483
KW 110 DATA 141,184,154,165,134,133,203,165,135,133,1547
TV 120 DATA 204,160,0,177,203,201,65,240,33,24,1307
GR 130 DATA 165,203,105,8,133,203,165,204,105,0,1291
FL 140 DATA 133,204,56,165,204,197,137,144,228,165,1633
OU 150 DATA 203,197,136,144,222,169,0,133,212,133,1549
DG 160 DATA 213,96,160,6,177,203,201,1,208,215,1480
WY 170 DATA 169,0,141,47,2,160,2,24,177,203,925
QI 180 DATA 101,140,133,112,200,177,203,101,141,133,1441
ZB 190 DATA 113,200,177,203,133,207,200,177,203,133,1746
EL 200 DATA 208,24,165,112,105,12,133,36,165,113,1073
DP 210 DATA 105,0,133,37,169,2,133,203,169,0,951
CN 220 DATA 133,204,56,165,203,233,1,133,205,165,1498
ZV 230 DATA 204,233,0,133,206,56,165,36,133,40,1206
CM 240 DATA 233,6,133,38,165,37,133,41,233,0,1019
UP 250 DATA 133,39,160,5,177,36,153,114,0,136,953
OU 260 DATA 16,248,162,5,181,114,149,212,202,16,1305
FY 270 DATA 249,160,5,177,38,153,224,0,136,16,1158
BK 280 DATA 248,32,45,155,16,47,160,5,177,38,923
SK 290 DATA 145,40,136,16,249,56,165,38,133,40,1018
RY 300 DATA 233,6,133,38,165,39,133,41,233,0,1021
JT 310 DATA 133,39,198,205,208,202,198,206,16,198,1603
RD 320 DATA 24,165,112,105,6,133,40,165,113,105,968
JG 330 DATA 0,133,41,160,5,185,114,0,145,40,823
WV 340 DATA 136,16,248,24,165,36,105,6,133,36,905
TB 350 DATA 165,37,105,0,133,37,230,203,208,2,1120
EV 360 DATA 230,204,56,165,204,197,208,176,3,76,1519
IT 370 DATA 122,154,56,165,203,197,207,176,3,76,1359
WA 380 DATA 122,154,56,165,207,233,1,133,212,165,1448
PL 390 DATA 208,233,0,133,213,169,34,141,47,2,1180
NL 400 DATA 96,56,165,212,197,224,240,33,8,104,1335
DW 410 DATA 41,1,72,165,212,16,12,165,224,16,924
JN 420 DATA 4,104,106,105,96,104,169,128,96,165,1078
EQ 430 DATA 224,16,2,104,96,104,73,1,76,66,762
SJ 440 DATA 155,162,1,248,56,181,212,213,224,208,1660
GI 450 DATA 13,232,224,6,208,244,165,212,48,10,1362
JV 460 DATA 169,0,216,96,176,246,165,212,48,246,1574
QQ 470 DATA 169,128,216,96,0,0,0,0,0,0,609

```



Program umožňuje uživatelům 8-bitových počítačů, kteří programují v jazyku BASIC, používat praktické okénkové menu. Podobný program sice už existuje, ale já jsem ho nevlastnil a po prostudování jeho popisu jsem se rozhodl napsat vlastní program, který lépe splní moje požadavky.

Při psaní programu jsem se snažil o snadné volání z BASICu, zabránění jakémkoliv kolizím mezi uchovávanými daty z okének a BASICovským programem, dostatečnou rychlosť. V zájmu úspory paměti nejsou kontrolovány vstupní parametry, takže je nutné program používat zodpovědně.

Umístění v paměti

Základní část programu je uložena v 6. stránce. Obsahuje všechny obslužné rutiny pro mazání a výpis okének. Tato část se volá prostřednictvím pomocné rutiny umístěné v řetězci o délce 56 byte. Úkolem této rutiny je pouze komunikace mezi BASICem a strojovým programem v 6. stránce.

Způsob volání

Pro vytvoření okénka s textem je nutné nejprve dimenzovat dvě řetězcové proměnné. První bude sloužit k rezervování paměti, do které bude program ukládat obsah části obrazové paměti, která bude přemazána vytvořeným okénkem. Její velikost je určena velikostí obsahu okénka. V druhé řetězcové proměnné bude umístěn text, který chceme v okénku vypsat. Program vypisuje texty do okénka od levého horního rohu. Pokud narazí v textu na značku SHIFT=, přejde ve výpisu textu na další řádek, který začíná za levým okrajem okénka. Text je ukončen inverzní značkou SHIFT=. Je nutné dbát na to, aby text nepřesahoval přes okraj okénka, neboť by po jeho zrušení přesahující část textu zůstala na obrazovce. Po dimenzování příslušných polí se příkazem USR volá adresa pomocné rutiny, za kterou následuje 6 parametrů: poloha levého horního rohu okénka X,Y, velikost okénka X,Y, adresa řetězce, do kterého se bude ukládat přepsaná oblast videostránky, adresa řetězce s vypisovaným textem. Takto se provádí otevírání okénka. Při jeho rušení je postup stejný, neuvádí se ale adresa řetězce s textem. Okénko je orámované a jeho velikost v ose X je o jednu větší než parametr. Na to je nutné myslit při dimenzování proměnné pro uchování obrazové paměti. Po otevření okénka program čeká na stisk klávesy. ASCII kód stisknuté klávesy uloží na adresu 0 a vrátí řízení BASICu. Jak již bylo uvedeno, program není ošetřen například proti otevírání nesmyslně malých či velkých okének, jejich chybnému umístění. To může mít za následek zhroucení systému.

Způsob využití si jistě najde každý sám. K pochopení ovládání pomůže ukázka, která je součástí výpisu programu.

Listing 1 - program PMWIND.BAS

Part : 1

BASIC



BASICové melodie aneb Jak potrápit sebe i BASIC

Většina programátorů by měla souhlasit s tvrzením, že dostat více ze zvukového chipu počítače Atari, zvláště při tvorbě hudby, je možné jen s použitím strojového jazyka. To je však pro začátečníky příliš těžké. Proto se uchyluje k používání jazyka BASIC, který je však daleko pomalejší a málo pružný, a proto nedovolí vytváření složitějších melodií.

To však neznamená, že BASIC je naprosto nezpůsobilý ke "komponování" hudby. Naopak, při jeho správném použití můžeme dosáhnout potěšujících výsledků. To však vyžaduje trošku programátorské péče a plánování. Můžete být ohromeni výsledkem své práce, jestiž se neobáváte několika duševních mozolů.

Pro příklad nemusíme chodit daleko. Problém malé rychlosti BASICu je asi každému znám. Při tvorbě melodií se dá tomuto problému předejít celkem snadno. Melodii "rozeberete" na co nejjednodušší části. Cím méně bude nuten BASIC přemýšlet, tím rychleji bude program vykonáván. To znamená, že se musíte vzdát několika "extra" věcí, které jinak můžete využívat při tvorbě tónů. Například omezíte využívání zvukových křivek při imitaci hudebních nástrojů. Uvidíte, že tato oběť umožní neobvyčejně zvýšit komplexnost hudebního skladání.

Další problém, který nemůžeme jen tak přejít, je fakt, že BASIC může vykonávat pouze jednu práci za určitý čas. Pokud chcete, aby hrála hudba ve vašem BASICovém programu, všechno ostatní musí stát. Pohyb hráčů, který jste tak těžce vytvořili animací, musí počkat, dokud nezazní z reproduktoru TV poslední tóny hudební skladby. Jediná možnost, jak se vyhnout tomuto úskalí, je použití VBI ke hraní melodie, a to vyžaduje - jak jste jistě uholí - strojový jazyk. (Pro pokročilejší doporučuji přílohu VII zpravodaje AK Praha: Přerušení při vertikálním zatemnění (VBI) - překlad série článků z amerického časopisu pro 8-bitové Atari ANALOG Computing.)

První krok

Než půjdeme dál, natypujte si prosím program MELODIE.BAS, jehož výpis najdete na konci tohoto článku. Použijte program TYPO II ke správnému přepsání datových řádků.

Program spusťte příkazem RUN; činnost programu přerušte stisknutím tlačítka RESET nebo kombinací klávesy BREAK a příkazu END.

Nezní to špatně na BASIC, viďte? Podívejte se nyní na řádek 60 a řekněte mi, co tam vidíte. Vypadá to jako časová smyčka, že? Dokonce i když je BASIC velmi pomalý, naš program potřebuje tak málo procesorového času, že v zájmu poslechu hudby jej musíme zpomalovat. Experimentujte s různými hodnotami v této smyčce. Jestliže chcete slyšet BASICovou "výpalovačku", kompletně zrušte časovou smyčku.

Jak jsem již řekl, program je rychlý, protože procesor zpracovává málo příkazů. Celá "hraci" sekce programu je pouze na řádcích 10-90. Tuto sekci však můžeme ještě více zredukovat. Řádky 10, 40, 80 a 90 používáme pro znovupoužití dat, což značně snižuje velikost paměti, kterou program vyžaduje. Samozřejmě bychom mohli udělat program jednodušším způsobem s použitím "lineárních" dat. To znamená, že program by četl hudební data od začátku do konce skladby bez znovupoužití některých částí dat. To je však náročné na paměť...

Jak tedy správně uspořádat část programu, která generuje hudbu?

Skrytá tajemství

Stejně jako hudební skladba i naše kompozice vytvořená programem obsahuje několik jedinečných částí. Právě tato jedinečnost zvyšuje nároky na programování hudby. Je jen na nás, abychom při komponování skladeb netrpěli bolením hlavy. Musíme proto zvolit vhodnou programovací metodu.

Z našeho programu vidíte, že k získání výsledného zvukového efektu nejdříve přečteme řádek dat. Tato data použijeme ke změně výstupu každého ze čtyř zvukových kanálů a potom se ve smyčce vrátíme zpět na čtení dalšího datového řádku.

Samozřejmě bychom mohli reprodukovat hudbu jen pomocí velkého množství příkazů SOUND. Každá instrukce by měla vlastní parametry, což by eliminovalo problém odlišné délky not (tentotého problém si rozebereme později). Potom bychom mohli jednoduše zapnout zvukový kanál a dělat něco jiného, než přijde čas změnit parametry zvuku vzhledem k jiné notě. Takto můžeme manipulovat s každým zvukovým kanálem zvlášť, bez použití struktury smyčky.

Ale to by bylo příliš náročné, nehledě na to, že tato metoda by vyžadovala více volné paměti, než potřebuje metoda smyčky. Jako programátoři bychom měli vždy hledat nejlepší řešení daného problému.

Kdykoliv budete mít část programu, která opakuje stejné instrukce znova a znova, nejlepším řešením je zde použití metody smyčky. V našem případě však použití této metody narazí

na problém, který musíme vyřešit. Co se asi stane, když chceme, aby se na výstupu jednoho kanálu objevila čtvrtková nota, na dalším osminová a na dalším půlová? Řešení? Co tak měnit všechny zvukové registry ve stejný moment, a to vždy při průchodu smyčkou? Obtížné, ano...

Trik

...ale snadno řešitelné.

Bod 1.: Jestliže používáme metodu smyčky, každá nota v naší melodii musí mít stejnou délku a pokud vyžadujeme další opakování smyčky, měníme vždy všechny čtyři hlasu.

Bod 2.: Noty, které vytvářejí naši hudbu, nemají stejnou délku. Proto musí být všechny čtyři hlasu rytmicky nezávislé.

Samořejmě vidíme, že body 1 a 2 nebude snadné dát dohromady. Pokud se tedy rozhodneme, že je účelné použít smyčku v našem programu, musíme změnit část naší teorie. Podívejme se, co můžeme dělat, abychom zahrnuli bod 2 do plánu programu. Předefinujme ho takto:

Bod 2.: Noty, které tvoří hudbu, se nevyskytují se stejnou délkou. Proto všechny čtyři hlasu musí být rytmicky nezávislé na našem sluchovém ústroji.

Co to pro nás znamená? Noty můžeme jednoduše skládat – např. čtvrtková nota může být složena ze dvou osminových not hraných bez pauzy za sebou. Podobně půlová nota jsou dvě čtvrtkové noty spojené dohromady.

Takže nám stačí najít v notové osnově notu nejkratší délky a potom přepsat všechny ostatní noty vzhledem k ní. Struktura smyčky našeho programu se tak stává lehce proveditelnou. Jinými slovy, jestliže hudba, kterou si přejeme reprodukovat na našem počítači Atari, se skládá z osminových, čtvrtkových a půlových not, musíme změnit každou čtvrtkovou notu na dvě osminové a každou půlovou na čtyři osminové noty. Počítač tak bude hrát sérii osminových not, což nám umožní používání metody smyčky a zabezpečuje plnou "kompatibilitu" bodů 1 a 2.

Data

Všechna data pro generování naší melodie najdeme ve výpisu 1 od řádku 1000. Platí zde několik zásad, což zjednoduší editování. Data jsou organizována přes takty; každý takt je vytvořen z osmi datových řádků. Každý hlas potřebuje pro generování zvuku dvě datové hodnoty. Proto každý datový řádek obsahuje osm čísel, dvě čísla pro každý ze čtyř hlasů. Není to shoda okolností, že každý takt má osm datových řádků. Při rozpisu skladby jsem našel osminové noty jako noty nejkratší délky. Na takt máme čtyři úhozy, každá úhoz má délku čtvrtkové noty. Tak kolik osminových not dostaneme na takt? (8)

Každý datový řádek tedy reprezentuje jednu osminovou notu pro každý hlas. První páár dat je použit pro SOUND0, druhý páár pro SOUND1, třetí páár pro SOUND2 a čtvrtý páár pro SOUND3. První hodnota každého páru je výška noty. Podívejte se na tabulku 1, kde najdete hodnoty potřebné k duplikování různých výšek not. Druhé číslo páru je hodnota pro krátkou pauzu mezi notami.

Můžeme to takto provést, protože dvě noty stejné výšky budou znít spolu, dokud je neoddělíme. Nemůžeme však mít pauzu po každé notě. Někdy potřebujeme vytvořit efekt slyšnutí not, např. dvou osminových not na jednu čtvrtkovou. Vzpomínáte si?

"Procházka" programem

Podívejme se krátce, jak je program sestaven. Podobně můžete vytvářet své vlastní verze.

- Řádek 110 -

Způsobuje přechod běhu programu na řádek 180 vždy po vyčerpání všech "hudebních" dat.

- Řádek 120 -

Čte (první) řádek dat.

- Řádek 130 -

Používá hodnoty načtených do proměnných P0...P3 ke krátkému vypnutí příslušného zvukového kanálu (pokud je hodnota proměnné rovna 0). To nám dovolí mít dvě po sobě jdoucí noty stejné výšky bez jejich "smíchání" do jedné noty. Hodnota proměnné je parametr hlasitosti pro instrukci SOUND. Pro pauzu zadájte 0. Pro pokračování bez pauzy zadáváme hodnotu rovnou parametru hlasitosti pro příslušnou instrukci SOUND na řádku 150.

- Řádek 140 -

Protože během druhé části skladby se používá i SOUND3, použijeme příkaz IF k indikaci jeho zapnutí. To nás ušetří před typováním zbytečných dat.

- Řádek 150 -

Zde používáme hodnoty výšky tónů (přečtené z dat) k zapnutí zvukových kanálů.

- Řádek 160 -

Tato zpožďovací smyčka nastavuje délku znění noty.

- Řádek 170 -

Jdi na začátek programu (řádek 120) a čti další řádek dat.

- Řádek 180 -

Když přečteme všechna data, instrukce TRAP na řádku 110 způsobí, že program zde bude pokračovat. Nastavíme proměnnou FLAG ke správné aktivaci hlasu 3, také nastavíme datový ukazatel na hodnotu první části melodii (vynechání úvodu) a novým příkazem TRAP zajistíme další běh programu. Potom se vracíme na řádek 120 k opětnému čtení dat.

- Řádek 190 -

Když přečteme datové hodnoty podruhé, nastavený příkaz TRAP na řádku 180 způsobí, že program zde bude pokračovat. Celý tento řádek je určen k inicializaci programu pro opakování celé melodie. Jediná cesta, jak přerušíte běh programu, je použití tlačítka BREAK nebo RESET. (Nebo, pokud jste "počítačový desperát", vypněte zdroj počítače.)

Ať žije hudba

A tak ji máte: jednoduchou programovou cestu ke generování hudebních skladeb v BASICu. Vezměte si nějakou melodii a zkuste to. Je to sice puntičářský proces, ale pokud jste trochu plní, zvládnete jej bez problémů a budete potěšeni výsledným efektem.

Tabulka 1: Hodnoty pro generování výšky not

vysoké C	29	střední C	121
B	31	B	128
A#	33	A#	136
A	35	A	144
G#	37	G#	153
G	40	G	162
F#	42	F#	173
F	45	F	182
E	47	E	193
D#	50	D#	204
D	53	D	217
C#	57	C#	230
C	60	C	243
B	64		
A#	68		
A	72		
G#	76		
G	81		
F#	85		
F	91		
E	96		
D#	102		
D	108		
C#	114		



Listing 1 - program MELODIE.BAS

Part : 1

```

WY 10 REM [REDACTED]
MF 20 REM [REDACTED] BASICally Melodic !
XQ 30 REM [REDACTED]
FR 40 REM - for all ATARI XL/XE
QP 50 REM (c) 1987
XW 60 REM by Clayton Walnum
UI 70 REM ANALOG Computing 5/1987
WJ 80 REM (o) 1989
DZ 90 REM by GIA Software
NC 110 TRAP 180
HI 120 READ F0,P0,F1,P1,F2,P2,F3,P3
B0 130 SOUND 0,F0,0,P0:SOUND 1,F1,10,P1:SOUND 2,F2,10,P2
OU 140 IF PASS=2 THEN SOUND 3,F3,10,6
BE 150 SOUND 0,F0,0,4:SOUND 1,F1,10,8:SOUND 2,F2,10,6
KM 160 FOR X=1 TO 50:NEXT X
MQ 170 GOTO 120
CG 180 PASS=2:RESTORE 1370:TRAP 190:GOTO 120
GM 190 RESTORE :PASS=1:SOUND 3,0,0,0:GOTO 110
VI 999 STOP
BQ 1000 REM MEASURE 1
YX 1010 DATA 50,4,0,0,0,0,0,0
ZV 1020 DATA 0,0,0,0,0,0,0,0
CJ 1030 DATA 1,4,0,0,0,0,0,0
AM 1040 DATA 1,0,0,0,0,0,0,0
ZJ 1050 DATA 50,4,0,0,0,0,0,0
XI 1060 DATA 50,0,0,0,0,0,0,0
CV 1070 DATA 1,4,0,0,0,0,0,0
AY 1080 DATA 1,0,0,0,0,0,0,0
DJ 1090 REM MEASURE 2
YW 1100 DATA 50,4,0,0,0,0,0,0
ZU 1110 DATA 0,0,0,0,0,0,0,0
CI 1120 DATA 1,4,0,0,0,0,0,0
AL 1130 DATA 1,0,0,0,0,0,0,0
ZI 1140 DATA 50,4,0,0,0,0,0,0
XH 1150 DATA 50,0,0,0,0,0,0,0
CU 1160 DATA 1,4,0,0,0,0,0,0
RX 1170 DATA 1,0,0,0,0,0,0,0
EA 1180 REM MEASURE 3
JL 1190 DATA 50,4,0,0,144,6,0,0
FA 1200 DATA 0,0,0,0,144,6,0,0
CH 1210 DATA 1,4,0,0,0,0,0,0
FR 1220 DATA 1,0,0,0,144,6,0,0
DH 1230 DATA 50,4,0,0,144,0,0,0
GU 1240 DATA 50,0,0,0,144,6,0,0
CT 1250 DATA 1,4,0,0,0,0,0,0
GC 1260 DATA 1,0,0,8,0,0,0,0
ER 1270 REM MEASURE 4
JK 1280 DATA 50,4,0,0,144,6,0,0
GB 1290 DATA 0,0,0,0,144,6,0,0
CG 1300 DATA 1,4,0,0,0,0,0,0
FQ 1310 DATA 1,0,0,0,144,6,0,0
DG 1320 DATA 50,4,0,0,144,0,0,0
GT 1330 DATA 50,0,0,0,144,6,0,0
CS 1340 DATA 1,4,0,0,0,0,0,0
GB 1350 DATA 1,0,0,8,0,0,0,0

```

Listing 1 - program MELODIE.BAS

Part : 2

```

FI 1360 REM MEASURE 5
JJ 1370 DATA 50,4,0,0,144,6,0,0
GA 1380 DATA 0,0,0,0,144,6,0,0
OG 1390 DATA 1,4,60,0,0,0,47,4
YW 1400 DATA 1,0,53,0,144,6,45,4
XJ 1410 DATA 50,4,47,0,144,0,40,4
EM 1420 DATA 50,0,53,0,144,6,45,4
NQ 1430 DATA 1,4,60,0,0,0,47,4
NS 1440 DATA 1,0,47,8,0,0,40,4
FZ 1450 REM MEASURE 6
JO 1460 DATA 50,4,47,8,144,6,40,4
BM 1470 DATA 0,0,47,8,144,6,40,4
OF 1480 DATA 1,4,60,0,0,0,47,4
ZX 1490 DATA 1,0,53,0,144,6,45,4
XI 1500 DATA 50,4,47,0,144,0,40,4
EL 1510 DATA 50,0,53,0,144,6,45,4
NP 1520 DATA 1,4,60,0,0,0,47,4
NR 1530 DATA 1,0,47,8,0,0,40,4
GQ 1540 REM MEASURE 7
SJ 1550 DATA 50,4,47,8,192,6,64,4
FU 1560 DATA 0,0,47,8,192,6,60,4
NS 1570 DATA 1,4,64,0,0,0,53,4
DA 1580 DATA 1,0,60,0,192,6,47,4
KB 1590 DATA 50,4,53,8,192,0,45,4
AD 1600 DATA 50,0,60,0,192,6,40,4
NE 1610 DATA 1,4,64,0,0,0,35,4
GG 1620 DATA 1,0,53,0,0,0,31,4
HH 1630 REM MEASURE 8
OH 1640 DATA 50,4,53,8,192,6,35,4
BW 1650 DATA 0,0,53,8,192,6,40,4
OR 1660 DATA 1,4,64,0,0,0,45,4
CZ 1670 DATA 1,0,60,0,192,6,47,4
DB 1680 DATA 50,4,53,0,192,0,53,4
DG 1690 DATA 50,0,60,0,192,6,60,4
DY 1700 DATA 1,4,64,0,0,0,54,4
LW 1710 DATA 1,0,53,0,0,0,54,4
HY 1720 REM MEASURE 9
OI 1730 DATA 50,4,53,8,182,6,72,4
HE 1740 DATA 0,0,53,8,182,6,64,4
LA 1750 DATA 1,4,72,0,0,0,60,4
AI 1760 DATA 1,0,64,0,182,6,53,3
EC 1770 DATA 50,4,60,0,182,0,47,4
HE 1780 DATA 50,0,64,0,182,6,53,4
LM 1790 DATA 1,4,72,0,0,0,60,4
LP 1800 DATA 1,0,60,0,0,0,47,4
LL 1810 REM MEASURE 10
PF 1820 DATA 50,4,60,8,182,6,47,4
GZ 1830 DATA 0,0,60,8,182,6,47,4
KZ 1840 DATA 1,4,72,0,0,0,60,4
BK 1850 DATA 1,0,64,0,182,6,53,4
EB 1860 DATA 50,4,60,0,182,0,47,4
HD 1870 DATA 50,0,64,0,182,6,53,4
QZ 1880 DATA 1,4,72,8,0,0,60,4
PC 1890 DATA 1,0,72,8,0,0,60,4
MD 1900 REM MEASURE 11

```

Listing 1 - program MELODIE.BAS

Part : 3

```

PL 1910 DATA 50,4,81,8,162,6,29,4
IC 1920 DATA 0,0,81,8,162,6,57,4
OO 1930 DATA 1,4,81,8,0,0,31,4
GQ 1940 DATA 1,0,81,8,162,6,64,4
GW 1950 DATA 50,4,81,8,162,0,35,4
LN 1960 DATA 50,0,81,8,162,6,72,4
PB 1970 DATA 1,4,81,8,0,0,40,4
RV 1980 DATA 1,0,81,8,0,0,81,4
NX 1990 REM MEASURE 12
NB 2000 DATA 50,4,91,8,162,6,45,4
FR 2010 DATA 0,0,91,8,162,6,91,4
VD 2020 DATA 1,4,91,8,0,0,47,4
LN 2030 DATA 1,0,91,8,162,6,96,4
GA 2040 DATA 50,4,81,8,162,0,53,4
RD 2050 DATA 50,0,91,8,162,6,108,4
QS 2060 DATA 1,4,91,8,0,0,60,4
MV 2070 DATA 1,0,91,8,0,0,121,4
NW 2080 REM MEASURE 13
CN 2090 DATA 50,4,96,8,192,6,0,0
VG 2100 DATA 0,0,96,8,192,6,0,0
LX 2110 DATA 1,4,96,8,0,0,0,0
VX 2120 DATA 1,0,96,8,192,6,0,0
WD 2130 DATA 50,4,96,8,192,0,0,0
ZW 2140 DATA 50,0,96,8,192,6,0,0
MJ 2150 DATA 1,4,96,8,0,0,0,0
KM 2160 DATA 1,0,96,8,0,0,0,0
OO 2170 REM MEASURE 14
YB 2180 DATA 50,4,96,8,192,6,76,4
PJ 2190 DATA 0,0,96,8,192,6,76,4
ZY 2200 DATA 1,4,96,8,0,0,76,4
OY 2210 DATA 1,0,96,8,192,6,76,4
RR 2220 DATA 50,4,96,8,192,0,76,4
VK 2230 DATA 50,0,96,8,192,6,76,4
AK 2240 DATA 1,4,96,8,0,0,76,4
YN 2250 DATA 1,0,96,8,0,0,76,4
PG 2260 REM MEASURE 15
TP 2270 DATA 50,4,76,8,192,6,64,4
LC 2280 DATA 0,0,76,8,192,6,64,4
WZ 2290 DATA 1,4,76,8,0,0,64,4
KR 2300 DATA 1,0,76,8,192,6,64,4
NF 2310 DATA 50,4,76,8,192,0,64,4
QY 2320 DATA 50,0,76,8,192,6,64,4
WJ 2330 DATA 1,4,76,8,0,0,64,4
UM 2340 DATA 1,0,76,8,0,0,64,4
PY 2350 REM MEASURE 16
PN 2360 DATA 50,4,64,8,192,6,53,4
HF 2370 DATA 0,0,64,8,192,6,53,4
TG 2380 DATA 1,4,64,8,0,0,53,4
HW 2390 DATA 1,0,64,8,192,6,53,4
JD 2400 DATA 50,4,64,8,192,0,53,4
MW 2410 DATA 50,0,64,8,192,6,53,4
SQ 2420 DATA 1,4,64,8,0,0,53,4
QT 2430 DATA 1,0,64,8,0,0,53,4
QQ 2440 REM MEASURE 17
PI 2450 DATA 50,4,53,8,192,6,45,4

```

130 XE



ANIMATION

Animace na počítači ATARI 130 XE

Počítače Atari 130 XE mají značnou převahu nad ostatními 8-bitovými typy firmy Atari. Důvodem je paměť 128 kB RAM, která poskytuje uživateli nemalé výhody, především jako RAMdisk. Jednou ze zajímavých možností tohoto počítače je animace s využitím rozšířené paměti 64 kB (extended memory). Pro animaci můžeme dokonce využít grafického modu s největší rozlišovací schopnosti, tj. grafického modu bez textového okénka GRAPHICS 8+16 (320 x 192 bodů). Obraz v této grafice zaujímá téměř 32 stránek paměti RAM (7680 bytů), proto v počítačích bez této rozšířené paměti (jako je Atari 800 XL/XE) lze současně umístit do RAM nejvýše čtyři obrázky v tomto režimu grafiky.

Proces animace vychází ze zásad uveřejněných v Atari zpravodaji Olomouc 1-2 1989. Technika využití rozšířené paměti RAM je zde vyřešena trochu jinak – dochází k přepínání nejen vyššího bytu adresy programu procesoru ANTIC, ale také banky paměti. Při sestavování programu je třeba mít na zřeteli takové rozmístění obrazů do jednotlivých bank rozšířené paměti, které umožní jejich přepínání pomocí jediné instrukce POKE. To znamená, že výsledného efektu docílíme buď přepínáním bank, nebo změnou adresy procesoru ANTIC. V opačném případě se budou obrázky překrývat a pohyb nebude plynulý.

Názorné použití popsané techniky nám přiblíží program ANIM130.BAS, který najdete ve výpisu 1 za tímto článkem. Nativujte program a zkонтrolujte jeho správnost pomocí programu TYPO II. Před použitím program uložte na záznamové médium – kazetu nebo disketu. Program spouštíme příkazem RUN. Pozor, použití tohoto příkazu ruší RAMdisk!

Uvedený program vykresluje 8 animačních fází (rotaci) pravidelného n-bokého hranolu ($n \geq 3$). Potom nastává přepínání jednotlivých obrázků, což vyslovává dojem otáčení tělesa. Pomocí tlačítka START a SELECT regulujeme rychlosť otáčení. Program lze zastavit stisknutím klávesy BREAK nebo tlačítka RESET.

Popis programu.

Rádek 10

- vkládání počtu stran pro základnu tělesa
(mnogoúhelník $n \geq 3$)

Rádek 20

- nastavení proměnných pro smyčku, ve které probíhá načítání jednotlivých obrazů

Rádky 30-40

- smyčka pro kreslení a animaci - v průběhu přípravy (vykreslování) jednotlivých obrazů je využíván podprogram na řádcích 80-100, v průběhu vlastní animace (rotace tělesa) potom využíváme podprogram na řádcích 50-70

Rádky 50-70

- podprogram pro regulaci rychlosti otáčení tělesa - stisknutí tlačítka START rotaci zpomalí, stisknutí tlačítka SELECT rotaci tělesa zrychlí

Rádky 80-100

- vykreslení těles - po ukončení této fáze program přejde na řádek 110

Rádek 110

- nastavení proměnných použitých ve smyčce na řádcích 30-40 k přepínání obrazů a nastavení RAMTOP na jeho normální velikost

Rádek 120

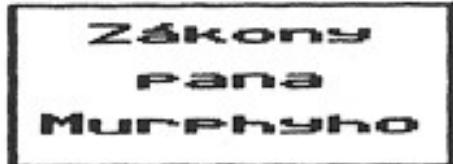
- počáteční nastavení rychlosti animace, skoku ve smyčce

Na závěr

Tímto programem jsem chtěli potěšit majitele počítače Atari 130 XE, kteří ještě neumí plně využívat možnosti, které jim rozšířená paměť dovoluje.

Když si program řádně prostudujete, zjistíte, jak je jednoduché využít rozšířenou paměť pro animaci. S programem můžete dále experimentovat. Vložte do programu nové proměnné, nebo stávající nahraďte jinými hodnotami a tím získáte zajímavé tvary a rozměry těles.

Pokud se Vám podaří program vylepšit nebo obměnit, napište nám. Zdařilé změny programu uveřejníme v některém z dalších čísel Zpravodaje roku 1990.

**Zákony vědeckého pokroku**

1. Užímkы jsou vždy početnější než pravidla.
2. Vždy se najdou užímkы ze stanovených Pravidel.
3. Až si konečně osvojíte všechny užímkы, už nikdo nebude schopen rozpoznat se na příslušná Pravidla, ke kterém se užímkы vztahují.

© 1989 by GIA

Listing 1 - program ANIM130.BAS

Part : 1

```

BV 1 REM
DQ 2 REM [ ANIMATION 130 XE ]
CI 3 REM
CB 4 REM (c) 1989 BAJTEK
JR 5 REM Upraveno:
XV 6 REM GIA Software
QY 7 REM Pozor!
WP 8 REM Tento program pracuje pouze
ZQ 9 REM s pocitacem ATARI 130 XE.
ND 10 GRAPHICS 0:?:?:?" Zadej pocet stran N=";:INPUT N:IF
    N<3 OR N>INT(N) THEN 10
DF 20 F=106:T=128:D=193:H=80:A=120:B=30:Q=35:P=160:KT=360/N:N=K
    T/16:DEG
JQ 30 FOR K=0 TO 8 STEP 8:E=4-K:POKE F,T-4*K:FOR J=0 TO 3:IF J
    THEN D=D+E
GS 40 POKE 54017,D:GOSUB H:NEXT J:NEXT K:POKE L,P:GOTO A
TQ 50 FOR I=1 TO N:IF PEEK(53279)=5 THEN IF N>0 THEN N=N-0.3
ZR 60 IF PEEK(53279)=6 THEN N=N+0.3
UT 70 NEXT I:RETURN
DD 80 GRAPHICS 8+16:COLOR 1:SETCOLOR 2,0,0
MV 90 FOR I=0 TO 360 STEP KT:X=P+A*COS(I+N):Y=Q+B*SIN(I+N):IF I
    THEN PLOT R,S:DRAWTO X,Y:DRAWTO X,Y+A:DRAWTO R,S+A
QZ 100 R=X:S=Y:NEXT I:FOR I=1 TO 300:NEXT I:N=N+KT/8:IF NKKT TH
    EN RETURN
SX 110 L=F:F=561:T=96:H=50:RETURN
NM 120 L=77:P=0:A=30:N=10:GOTO R

```

Listing 1 - program MELODIE.BAS

Part : 4

```

HB 2460 DATA 0,0,53,8,192,6,45,4
TA 2470 DATA 1,4,53,8,0,0,45,4
HS 2480 DATA 1,0,53,8,192,6,45,4
KA 2490 DATA 50,4,53,8,192,0,45,4
MR 2500 DATA 50,0,53,8,192,6,45,4
SK 2510 DATA 1,4,53,8,0,0,45,4
QN 2520 DATA 1,0,53,8,0,0,45,4
RI 2530 REM MEASURE 18
PH 2540 DATA 50,4,53,8,192,6,45,4
HA 2550 DATA 0,0,53,8,192,6,45,4
SZ 2560 DATA 1,4,53,8,0,0,45,4
HR 2570 DATA 1,0,53,8,192,6,45,4
JZ 2580 DATA 50,4,53,8,192,0,45,4
NS 2590 DATA 50,0,53,8,192,6,45,4
SJ 2600 DATA 1,4,53,8,0,0,45,4
QM 2610 DATA 1,0,53,8,0,0,45,4

```

Rozhodl jsem se svými dvěma malými programy přispět do Vaší rubriky MINI SOFTWARE 8-bitových počítačů ATARI XL/XE.

První program - výpis tabulky zařízení HATABS - pomůže každému programátorovi vypsat vstupní body každého ATARI zařízení. Po stisknutí písmene odpovídajícího zařízení se nám objeví vektory rutin OPEN, CLOSE, GETBYTE, PUTBYTE, STATUS, SPECIAL, INITIAL, to vše v desítkové a šestnáctkové soustavě.

Druhý program - 256 barev v textovém modu 9/0. Každého jistě zaujme, že moje řešení nezabírá ani 1 kB. Obrazovka je složena z načtených znaků; šířku obrazovky jsem upravil na dostatečných 32 znaků, takže výsledná obrazová paměť také nepřesahuje 1 kB. Dále jsem využil přerušení DL, které je uloženo na 6. stránce paměti. Přerušení DL obstarává změnu barvy po řádcích.

Autor programů a jejich popisu.

Ivo Cabrádek

Listing MiSe - program HATABS.BAS

Part : 1

```
TU 10 REM tab.HATABS/I.C.89
ZB 20 DIM HEX$(16),H$(4):HEX$="0123456789ABCDEF"
SA 30 GRAPHICS 0:POKE 710,0:POKE 82,2:POKE 201,8
CL 40 ? "TAB.HATABS:($31A-$33D)":A=794:? " ODSLZNE PROGRAMY"
MY 50 ? CHR$(PEEK(A));":":C=PEEK(A+1)+PEEK(A+2)*256:GOSUB 1010
:A=A+3:IF A<830 THEN 50
HB 60 ?:? " ZARIZENI":":GOSUB 1000
KU 70 FOR A=794 TO 809 STEP 3
TQ 80 IF PEEK(A)=K THEN 100
AS 90 NEXT A:GOTO 30
DK 100 A=PEEK(A+1)+PEEK(A+2)*256
KI 110 FOR B=A TO A+12 STEP 2:? ,:C=PEEK(B)+PEEK(B+1)*256-1:GOS
UB 1010:NEXT B
KQ 120 ? "1111111OPEN":? "CLOSE":? "GET B.":? "PUT B.":? "STATU
S":? "SPEC.":? "INIC.":GOSUB 1000:RUN
CR 1000 REM key
ZQ 1001 OPEN #1,4,0,"K":? GET #1,K:? CHR$(K):CLOSE #1:RETURN
SO 1010 REM dec-hex
BR 1020 H$="":Q=INT(C/256):H=INT(Q/16):H$(1)=HEX$(H+1,H+1):H=Q-
H*16:H$(2)=HEX$(H+1,H+1)
OI 1030 Q=C-Q*256:H=INT(Q/16):H$(3)=HEX$(H+1,H+1):H=Q-H*16:H$(4
)=HEX$(H+1,H+1)
PG 1040 ? C,"$":H$:RETURN
```

Listing MiSe - program C256.BAS

Part : 1

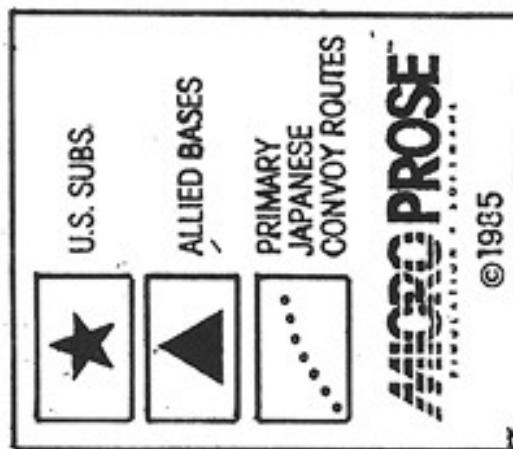
```
XP 10 REM GR.9/0
BR 20 GRAPHICS 0:POKE 559,0:POKE 752,1:POKE 82,0:POKE 83,39:CH=
37888:POKE 756,CH/256:POKE 623,64:POKE 712,0:Z=0
AH 30 POSITION 32,0
IJ 40 0=Z*8+CH:FOR Q=0 TO 6:POKE 0+Q,Z*17:POKE Q+0+128,Z:NEXT Q
:POKE 0+7,0:POKE 0+135,0:Z=Z+1:IF Z<16 THEN 40
JC 50 FOR Q=32 TO 47:? CHR$(Q+16);CHR$(Q);:NEXT Q:Z=Z+1:IF Z<32
THEN 50
LT 60 DATA 72,165,61,24,105,16,133,61,141,26,208,201,240,208,4,
169,0,133,61,104,64
GA 65 0=PEEK(560)+256*PEEK(561)
MS 70 FOR Q=0 TO 20:READ B:POKE Q+1536,B:IF Q>5 THEN POKE Q+0,1
30
JV 80 NEXT Q:POKE 61,0:POKE 512,0:POKE 513,6:POKE 54286,192:POK
E 559,33
UK 90 GOTO 90
```



MONDAY



THE PACIFIC OCEAN





SILENT SERVICE

SILENT SERVICE
 (The Submarine Simulation)
 by SID MEIER
 MICROPROSE (c) 1985

SILENT SERVICE je jedním z nejlepších simulačních programů pro osmibitové počítače ATARI. Díky tomuto programu se může každý stát velitelem ponorky za druhé světové války, při čemž jsou zachovány veškeré historické souvislosti.

Po nahrání programu se objeví následující menu:

1. TORPEDO/GUN PRACTICE (výcvik torpédrování a střelby z děla)
 Tento scénář tě umístí na americkou základnu na ostrově MIDWAY, kde jsou zakotveny čtyři staré nákladní lodě jako cvičné cíle
2. CONVOY ACTION (konvojové akce)
 Zde se již jedná o skutečné akce konvojových přepadů za druhé světové války.
3. WAR PATROL (válečné hlídkování)
 V tomto scénáři velíš celé hlídkové službě, počínaje ponorkovými základnami MIDWAY, BRISBANE nebo FREMANTLE (PERTH), pokračuje řadou konvojových akcí a návratem na základnu končí

TARGET IDENTIFICATION PRACTICE (výcvik rozpoznání cíle)

Pouze u scénáře číslo 3. Rozpoznání cíle je jedna z životně důležitých funkcí, kterou musí ovládat každý kapitán ponorky. Prohlédni si požadovanou loď (např. japonský torpédoborec - typ I /destroyer/) v tomto manuálu. Urči, která za čtyř lodních siluet zobrazených na obrazovce se podobá siluetě v manuálu. Zadej číslo této siluety (1,2,3 nebo 4). Jestliže správně rozpoznáš loď, smíš nastoupit obtížnou službu v Pacifiku. V případě, že odpovíš špatně, budeš pokračovat ve výcviku torpédrování a střelby na základně MIDWAY.

SKILL LEVELS (úrovně dovednosti)

Voli se stiskem kláves 1,2,3 nebo 4 pro volbu úrovně:

1. MIDSHIPMAN - pro začátečníky
2. LIEUTNANT - pro pokročilejší
3. COMMANDER - historicky přesná úroveň
4. CAPTAIN - pro experts

REALITY LEVELS (úrovně reality)

Pro volbu úrovně použij joy-stick nebo klávesy "-", "=" pro přemístění hvězdičky a spoušť nebo OPTION pro překlopení indikátoru YES/NO (ANO/NE).

1. LIMITED VISIBILITY (omezená viditelnost)

Lodě, které jsou mimo dosah radaru sonaru, nejsou zobrazeny na mapě. Nepřátelské lodě, které byly zjištěny, ale dostaly se z dosahu, budou blikat na jejich poslední známé pozici.

2. CONVOY ZIG-ZAGS (kličkující konvoj)

Zvolený nepřátelský konvoj bude kličkovat (měnit kurs) v pravidelných intervalech. Pokud tato úroveň nebude navolena, nákladní lodě poplují přímo vpřed, dokud na ně nebude zaútočeno torpédy nebo dokud se nesetkají se zemí.

3. DUD TORPEDOES (zmetková torpéda)

Některé vaše torpédo nemusí vybuchnout (zvláště během let 1942-1943). Zmetková torpéda mohou zasáhnout nepřitele, ale nevybuchnou. Pouze uvidíte šplouchnutí vody.

4. PORT REPAIRS ONLY (oprava pouze na základně)

Zničený člen hlavního vybavení nemůže být opraven.

5. EXPERT DESTROYERS (torpédoborce se zkušenou obsluhou)

Určité nepřátelské konvoje budou doprovázeny těmito

torpédoborci. Tyto eskorty jsou odolnější a mají lépe vycvičené sonarové operátory.

6. CONVOY SEARCH (hledání konvoje)

Konvoj se neobjeví vždy v dosahu radaru, a proto jej musíš hledat. Vzdálené konvoje se dají nejlépe zpozorovat prohlížením celého horizontu s použitím periskopu.

7. ANGLE-ON-BOW INPUT (zadávání úhlu přídě)

Počítač nebude už nadále vypočítávat úhel přídě. Musíš zadat úhel přídě sám na zakladě pozorování periskopem. Doporučuje se pro zkušené hráče.

DIFFICULTY LEVELS (úrovně obtížnosti)

Celková úroveň obtížnosti v rozsahu 1-9 je dána kombinací zvolené úrovně dovednosti a úrovně reality. Tento faktor a potopená tonáž určí tvé zařazení v tzv. "Dvoraně slávy ponorkových kapitánů" na závěr tvého úkolu.

Po navolení úrovně reality a dovednosti můžeš začít hrát

TERMINOLOGIE

PORT - levobok

STARBOARD - pravobok

BEARING - směr, kterým se díváš

HEADING - směr, kterým pluje ponorka

CONTROLS (ovládací prvky)

Všechny následující ovládací prvky se dají volit z klávesnice a jsou uvedeny ve stati POVELY Z KLÁVESNICE na konci manuálu.

1. SUBMARINE CONTROLS (ovládání ponorky)

VYSOUVÁNÍ A ZASOUVÁNÍ PERISKOPU

Provádí se pomocí klávesy P. Tento povel také nastaví vizuální kurs (BEARING) stejně jako kurs ponorky (HEADING). Maximální hloubka pro použití periskopu je 44 stop (pouze ve dne).

PLYN (RYCHLOST)

Nastavuje se pomocí kláves 0 až 4.

0 - zastaveno

1 - 1/3 maximální rychlosti

2 - 2/3 maximální rychlosti

3 - maximální rychlosť

4 - přídavná rychlosť

ZPĚTNÝ BEH

Nastavuje se pomocí klávesy R. Pro změnu rychlosti při zpětném běhu nastav plyn, pak zmáčkní znova R. Uvědom si, že účinek směrového kormidla je opačný než směr normálního pohybu.

PONORENÍ

Provádí se pohybem joysticku dolů (směrem k sobě). Když dosáhneš požadované hloubky, zrušíš tento povel tak, že dás joystick nahoru (od sebe). Ve scénářích před srpnem 1943 je maximální bezpečná hloubka 300 stop, po tomto datu je to 425 stop.

VYNORENÍ

Provedeš pohybem joysticku nahoru. Když dosáhneš požadované hloubky, zrušíš tento povel tak, že dás joystick dolů.

SMĚROVÉ KORMIDLO

Vlevo - joystick vlevo.
Vpravo - joystick vpravo.

Plná výchylka kormidla - joystick 2-krát požadovaným směrem.
Konec zatáčení - joystick na opačnou stranu.

ZRУSENÍ

Pro zrušení všech otáčecích a ponořovacích povelů stiskni klávesu RETURN.

VYPUSTENÍ NOUZOVÝCH NÁDRŽÍ

Stiskni klávesu SHIFT+E, tím se zastaví jinak smrtelně nebezpečný sestup a ponorka se obvykle vynoří. Můžeš použít pouze 1-krát za hru.

2. COMBAT CONTROLS (bojové ovládací prvky)

Jsou dostupné, především když je periskop v bojové poloze.

IDENTIFIKACE CÍLE

Po zaměření cíle v záměrném kříži (kříž zesvětlá) stiskni klávesu I pro jeho identifikaci.

ODPAL TORPED

Stiskni klávesu T. Torpéda na přídi nebo na zádi budou zvolena automaticky podle polohy cíle vůči ponorce (před či za ponorkou). Uvědom si, že současně mohou být v činnosti 4 torpéda a dělostřelecké granáty (celkově). Jestliže pátá výstřel přijde dřív, než přijde první, pak první skončí předčasně.

PALUBNÍ DELO

Ráže 4 palce – odpálení pomocí klávesy B. Střílet je možno pouze na hladině. Dostřel je nastaven automaticky na cíl, který je v zaměřovači.

ZVÝSENÍ DOSTRELU DĚLA

Pomocí klávesy > se zvýší dostřel děla o 25 yardů. Každé další stisknutí přidá dalších 25 yardů. Používá se, když se cíl pohybuje od tebe.

SNÍZENÍ DOSTRELU DĚLA

Pomocí klávesy < se sníží dostřel o 25 yardů. Používá se při pohybu cíle směrem k tobě.

OTÁCENÍ VÝHLEDU V PERISKOPU

Vlevo – joystick vlevo, vpravo – joystick vpravo. Pro zvýšení rychlosti otáčení drž spoušť joysticku.

ZADÁVÁNÍ ZÁMERNEHO ÚHLU

Pokud hraješ tuto úroveň reality (ANGLE-ON-BOW INPUT), musíš zadat zámerný úhel v rozsahu -180 až +180 stupňů předtím, než odpálíš torpédo na cíl. Před zadáváním stiskni klávesu A. Hodnota úhlu se zadává pomocí joysticku. Vlevo – záporný úhel, vpravo – kladný úhel. Po dosažení zvoleného úhlu zmáčkní tlačítko FIRE (spoušť).

ODSTRANĚNÍ TROSEK

Provádí se pomocí klávesy SHIFT+/. Tím se odstraní trosky a olej. Začnou stoupat na hladinu a simulují potopení ponorky. Používá se pro oklamání nepřítele. Lze použít pouze 1-krát za hru.

3. RIZENÍ ČASU A MĚRÍTEK

WAIT (pauza)

Potřebuješ-li na okamžik přerušit hru, stiskni klávesu W. Pokračovat ve hře budeš po stisku libovolné klávesy.

FAST (zrychlení času)

Pomocí této funkce můžeš urychlit například nudné stíhání nepřátelského konvoje. Stisknutí klávesy F způsobí zrychlení času 2-krát. Maximální zrychlení je 32-krát větší než reálný čas (tzn. tři stisky za sebou).

NORMAL TIME (normální čas)

Návrat do reálného časového měřítka provedeš stisknutím klávesy N. Automaticky se vrátíš do normálního času v případě, že tě odhalí nepřítel, nebo jakmile odpálíš torpédo.

3. VÁLEČNÉ HLÍDKOVÁNÍ

VELITELSKÁ VĚŽ

Použij joystick pro přemístění na požadované stanoviště a zmačknici FIRE.

STRED - periskop (USE SCOPE)
 NAHORE - můstek (GO TO BRIDGE)
 DOLO - hlášení o poškození (CHECK DAMAGE)
 VLEVO - přístrojová deska (VIEW GAUGES)
 VPRAVO - mapy (STUDY MAPS)

Na velitelskou věž se můžeš vrátit z kteréhokoliv místa, kromě můstku (BRIDGE), stisknutím tlačítka FIRE nebo SPACE BAR (mezerník). Když jsi ve velitelské věži, simulace je zastavena. Všimni si, že některé volby jsou nedostupné za určitých podmínek, například můstek, je-li pod vodou (BRIDGE UNDER WATER). Z hlídkové věže můžeš také zvolit dvě speciální funkce:

POLOHA VLEVO DOLE

1. Konec hry (v případě, že hrájeme TORPEDO/GUN PRACTICE nebo CONVOY ACTION)
2. Pokračování v hlídkování (CONTINUE PATROL) v případě, že jsme zvolili WAR PATROL. V hlídkování nemůžeš pokračovat, jsí-li pronásledován nepřátelskou eskortou, máš-li v činnosti torpéda, případně potápíš-li se nepřátelská loď.

POLOHA VPRAVO DOLE

- lodní deník (QUARTERMASTERS LOG). Jsou v něm zapsány všechny výsledky dosud dosažené při hlídkování.

BOJOVÉ STANOVÍSTE

NAVIGAČNÍ MAPA PRO VÁLEČNÉ HLÍDKOVÁNÍ.

Pokud zvolíš válečné hlídkování (WAR PATROL), začínáš touto scénou, která zobrazuje mapu západního Pacifiku. Nic ti nebrání v pozorování kteréhokoliv místa na mapě. Tvá ponorka je malá černá bod poblíž startovního přístavu FREMANTLE (PERTH), BRISBANE nebo MIDWAY. Hlídková scéna simuluje čas potřebný k dosažení nepřítele v kontrolovaných vodách a současně simuluje hlídkovou činnost (typická hlídka trvala až dva měsíce).

POHYB PO MAPĚ

Pomocí joysticku určuj směr plavby tvé ponorky (černá tečka). Čas běží při hlídkování velmi rychle a simuluje skutečnou dobu a rychlosť pro dosažení cíle na moři. Oceán se mění od světle modré (den) po tmavě modrou (noc).

VYHLEDÁNÍ NEPRÍTELE

Když oceán zčervená, vypátral jsi konvoj. Stiskem spouště nebo tlačítka START přejdeš z hlídkování do boje. Všimni si, že nepřátelské lodě se nalézají především v okolí vyznačených konvojových tras (viz mapy) a podél pobřeží. Velké tankery a transportní konvoje plují pravděpodobně poblíž japonských ostrovů.

NÁVRAT DOMO

Ponorkové základny jsou označeny na mapě bílým bodem. Když dosáhneš základny, oceán zezeelená a ty se můžeš vrátit do přístavu. Zmáčkní spoušť a uvidíš své ohodnocení.
DŮLEŽITÉ UPOMÍNKY - na této scéně nelze použít žádné jiné ovládací prvky. Jediná možnost je oceán opustit.

MAPA

Zobrazují se na ní informace od navišátoru a sledovací skupiny. Mapa, vizuální porovnávání, radar a sonar jsou kombinovány tak, aby ti ukázaly umístění tvé ponorky, torpéda a všech známých nepřátelských lodí.

Cerná tečka - tvá ponorka a torpéda.

Bílá tečka - nepřátelské lodě.

Zelená plocha - pevnina a ostrovy.

POZNÁMKA: nezjištěné lodě se na mapě neobjeví (mimo oblast, kterou jsi prohledal). Tvé pozorovatelnny nejsou vždy spolehlivé, a proto je rozumné opustit toto stanoviště a prohlédnout si horizont sám, pomocí periskopu, nebo z můstku. Pokud nepřátelská loď zmizí z dohledu, tečka bude blikat na poslední známé pozici.

LUPA - klávesa Z zvětšuje měřítko mapy
klávesa X zmenšuje měřítko mapy

ÚROVNE MĚŘÍTEK

1. Počáteční mapa ukazuje celý západní Pacifik.
2. Mapa hlídkové oblasti představuje plochu 500 x 300 mil.
3. Navigační mapa 60 x 40 mil.
4. Mapa oblasti útoku 8 x 5 mil. Lodě jsou zobrazeny s malými ocásky, které označují směr pohybu lodí.

DOSTUPNÉ OVLÁDACÍ PRVKY: Všechny ponorkové, časové a měřítkové prvky + odstranění trosek. V případě, že jsou v činnosti torpéda, lze čas urychlit pouze 2x.

BOJOVÉ STANOVÍSTE - PERISKOP

Tato scéna zobrazuje pohled přes útočný periskop během dne, šera nebo svítání a pohled z optického zaměřovače umístěného na můstku v noci. Je vidět zvětšený obraz viditelných cílů, nebo země. Tato scéna se může navolit, když je ponorka na hladině, nebo v hloubce maximálně 44 stop (pod hladinou je potřeba nejprve vysunout periskop), a to pouze ve dne, protože v noci periskop nepropustí dost světla.
OVLÁDÁNÍ: všechny bojové funkce, pauza, zrychlení času a normální čas, odstranění trosek i ovládání ponorky.

BOJOVÉ STANOVÍSTE - MОСТЕК

OVLÁDÁNÍ: otáčení vlevo nebo vpravo, palubní dělo + ovládání dostřelu.

POZNAMKA: Řízení palby lodního děla z můstku se nedoporučuje. Raději použij stanoviště periskopu.

BEARING (vizuální kurs) je směr, kterým se díváš, vyjádřený v kompasových stupních.

- 000 - sever
- 045 - severovýchod
- 090 - východ
- 135 - jihozápad
- 180 - jih
- 225 - jihozápad
- 270 - západ
- 315 - severozápad

POČÍTAČ TORPEDOVÝCH DAT

Jakmile záměrný kříž zbělá, je počítač aktivován a zobrazí parametry cíle:

- vzdálenost cíle (TARGET RANGE)
- rychlosť cíle (TARGET SPEED)
- úhel přídě (ANGLE ON BOW)
- gyroskopický úhel (LEAD ANGLE)
- kurs cíle (TARGET COURSE)

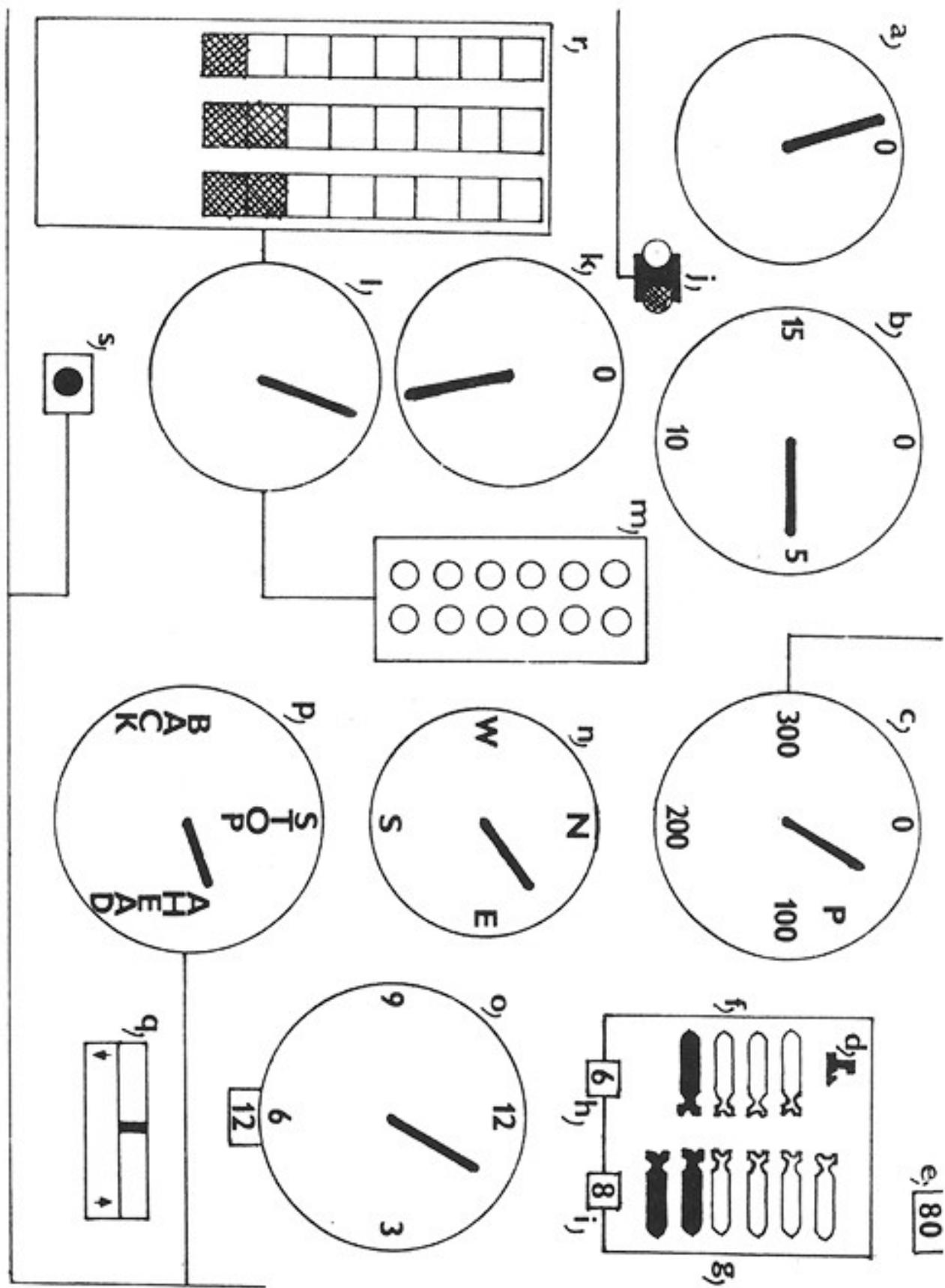
Kurs cíle není dostupný v případě, že máš navolenou úroveň reality - zadávání úhlu přídě (=ANGLE ON BOW INPUT). Můžeš odpálit torpédo stiskem klávesy T, vypálit z děla stiskem klávesy G nebo požadovat informace o cílu od identifikační skupiny klávesou I.

PŘISTROJE

(použij obrazovou přílohu přístrojů)

a, Ukazatel stavu baterií - znázorňuje zbývající kapacitu baterií. Baterie se používají pro plavbu pod vodou a nabíjejí se na hladině. Plně nabité baterie vydrží jednu hodinu rychlé jízdy pod vodou a pět až šest hodin jízdy pomalé. Ručička na přístroji se při vybití baterií pohybuje proti směru hodinových ručiček k 0 (obr. baterie nabité na maximum).

2 SPEED: 5 DEPTH: 42 HEADING: 061



Pohled na přístroje

b, Rychloměr - maximální rychlosť je 20 uzlů na hladině a 10 uzlů pod hladinou v případě, že nejsou poškozeny motory (podle přílohy se ponorka pohybuje rychlosťí 5 uzlů).

c, Hloubkoměr - hloubka je udávána ve stopách (podle přílohy je hloubka 42 stop).

d, Indikátor periskopu - je-li bílá, pak je periskop vytážen, černý znamená - periskop zatažen (kontrolka v příloze hlásí, že periskop je zatažen).

e, Číslo indikuje počet zbývajících dělostřeleckých nábojů (80 - maximální počet nábojů).

f, Indikátor připravenosti zadních torpédonosných skupin. Skupina 4 kontrolky ve tvaru torpéda ukazuje počet plných a prázdných torpédometů. Bílá kontrolka - torpédo připraveno k odpalování, černá kontrolka znamená prázdný torpédomet (podle přílohy jsou 3 torpéda připravena k odpálení a jeden torpédomet je prázdný).

g, Indikátor připravenosti předních torpédonosných skupin. Podobně jako u zadních torpédonosných skupin (podle přílohy jsou 4 torpéda připravena v torpédometech a 2 přední torpédomety jsou prázdné).

h, Číslo ukazuje počet zbývajících zadních torpédonosných skupin mimo ty, které jsou v torpédometech. Nabíjení torpédonosné skupiny se děje automaticky a trvá asi 10 herních minut pro jedno torpédo (dle přílohy zbývá ještě 6 zadních torpédonosných skupin + 3 v torpédometech).

i, Totéž jako v předcházejícím bodě, ale pro přední torpédonosné skupiny (dle přílohy zbývá ještě 8 předních torpédonosných skupin + 4 v torpédometech).

j, Levá kontrolka signalizuje dobíjení baterií a pravá kontrolka ukazuje odběr elektrické energie z baterií (v příloze svítí pravá kontrolka, to znamená odběr elektřiny).

k, Hloubka pod kýlem - vzdálenost ponorky od mořského dna. Údaj 0 znamená, že ponorka je na dně. Maximální údaj je 500 stop.

l, Teploměr - teplota okolní vody.

m, Vánoční stromeček - tablo kontrolky indikujících stav všech ponorkových uzávěrů. Uzávěry se zavírají automaticky při povolení k ponorování (v příloze jsou všechny uzávěry zavřeny).

n, Kompas - udává směr plavby ponorky (HEADING), (v příloze je to 61 stupňů).

o, Hodiny - ručička ukazuje minuty a číslo pod hodinami danou hodinu. V Pacifiku se stmívá mezi devatenáctou a dvacátou hodinou a svítá okolo páté až šesté hodiny ranní (hodiny v příloze ukazují 12 hodin a 5 minut).

p, Indikátor nastavení rychlosti a směru pohybu.

BACK - zpětný pohyb

STOP - zastavení

AHEAD - pohyb vpřed

Ručička ukazuje rovněž nastavení plynů (ponorka se podle přílohy pohybuje vpřed rychlostí 5 uzlů se zařazenou druhou rychlostí).

a, Indikátor ponoru - čárka ukazuje, zda se ponorka vynořuje (čárka vlevo), nebo ponoruje (čárka vpravo). (v příloze ponorka setrvává ve stejné hloubce)

r, Palivoměr - tři svislé sloupce ukazují množství paliva ve třech hlavních nádržích. Palivo plave na hladině vody. Sloupce ukazují množství paliva (bílá) a vody (černá) v každé nádrži. Plné nádrže vystačí na 50 až 60 dní plavby.

s, Tato kontrolka se rozsvítí, jestliže teplota okolní vody poklesne pod hranici teplotního gradientu, což znesnadňuje nepříteli zaměření ponorky.

STANOVÍSTE HLÁSENI O POSKOZENÍ

BOW TORPEDO DAMAGE - poškozený přední torpédomet.

AFT TORPEDO DAMAGE - poškozený zadní torpédomet.

PERISCOPE DAMAGE - poškozený periskop.

DIVE PLANE DAMAGE - poškozena hloubková kormidla (lze se ponořovat a vynořovat pouze poloviční rychlostí).

FUEL LEAKING - únik paliva. Spotřeba paliva se zdvojnásobí a toto palivo stoupající k hladině může upozornit nepřátelské torpédoborce na ponorku.

ENGINE DAMAGE - poškození motorů. Jsou poškozené hlavní dieselové motory. Rychlosť na hladině se sníží o polovinu.

MACHINERY DAMAGE - poškození strojů. Vnitřní čerpadla a motory jsou poškozeny. Zvětšený hluk usnadňuje nepříteli vypátrání ponorky pomocí sonaru.

BATTERY DAMAGE - poškození baterií. Baterie se vyčerpávají dvakrát rychleji.

Pokud není navolena úroveň reality PORT REPAIRS ONLY (opravy pouze v přístavu), provádí se opravy automaticky.

Jestliže ponorka nabírá vodu, je rychlosť nabíráni indikována v salonech za sekundu (GPS). To způsobuje, že ponorka klesá, i když jsou hloubková kormidla nastavena opačně. Tato informace je uvedena vpravo nahoře na této scéně.

DIAGRAM RIZENÍ A STAV PONORKY

Dolních několik řádek ve většině bojových stanovišť obsahuje diagram řízení a stav ponorky (viz spodní řádek na přístrojové desce v příloze). Zcela dole je zobrazena zvolená rychlosť ponorky (2 - tzn. 2/3 rychlosti), poloha směrového

kormidla (kormidlo natočeno vpravo), šípka nahorá nebo dolů označuje buď stoupání, nebo klesání ponorky (není-li šípka vidět, potom ponorka udržuje stálou hloubku), rychlosť v uzlech (SPEED: 5), hloubka ve stopách (DEPTH: 42= a směr plavby (HEADING: 061). Horní (dialogový) rámek slouží jako informace o stavu ponorky (v příloze je dialogový rámek prázdný).

ZPRÁVY A ZVUK

Dostáváš zprávy kdykoliv od různých členů posádky. Povely pro směrové kormidlo, plyn a periskop budou potvrzovány v dialogovém rámečku. Budeš také slyšet zvuk vlastního motoru, blízkých lodí a torpédu. Vedle toho jsou také v dialogovém rámečku zobrazovány zprávy se zvláštním významem.

SONAR REPORTS DESTROYERS CLOSING (ping zvuk) – operátor sonaru hlásí, že ponorka byla objevena nepřátelským sonarem.

SONAR REPORTS DEPTH CHARGES DROPPED (zvuk šplouchnutí) – operátor sonaru hlásí, že torpédoborec nad ponorkou vypustil hlubinné miny.

DEPTH CHARGES EXPLODING (zvuk výbuchu) – výbuch hlubinných min.

LOOKOUTS REPORT DESTROYERS FIRING (zvuk výstřelu) – pozorovatelé na můstku hlásí, že nepřátelské torpédoborce jsou na dostřel a střílí na ponorku.

SHELL HIT! SUB DAMAGE (hvízdavý zvuk výbuchu) – tvoje ponorka byla zasažena nábojem z nepřátelského torpédoborce a poškozena.

BOW (AFT) TORPEDO FIRED! 135 TRACK (zvuk motoru torpéda) – start předního (zadního) torpéda pod úhlem 135 stupňů.

DECK GUN FIRED (zvuk palubního děla) – výstrel z palubního děla.

SONAR REPORTS DISTANT EXPLOSIONS (zvuk vzdáleného výbuchu) – operátor sonaru hlásí výbuch torpéda nebo granátu.

WARNING: TEST DEPTH EXCED (houkání) – výstraha. Je překročena maximální povolená hloubka (ponorka začíná nabírat vodu – zkontroluj scénu hlášení o poškození).

WE HAVE RUN AGROUND (skřípavý zvuk) – najeli jsme na mělčinu.

REPAIRS COMPLETED – opravy dokončeny.

BLOW EMERGENCY TANK (zvuk poplachu) – havarijní nádrže vypuštěny.

RAMMED BY ENEMY SHIP (skřípavý zvuk) – nepřátelská loď najela na ponorku a začínáš se potápět. Obvykle to bývá osudné.

CAS

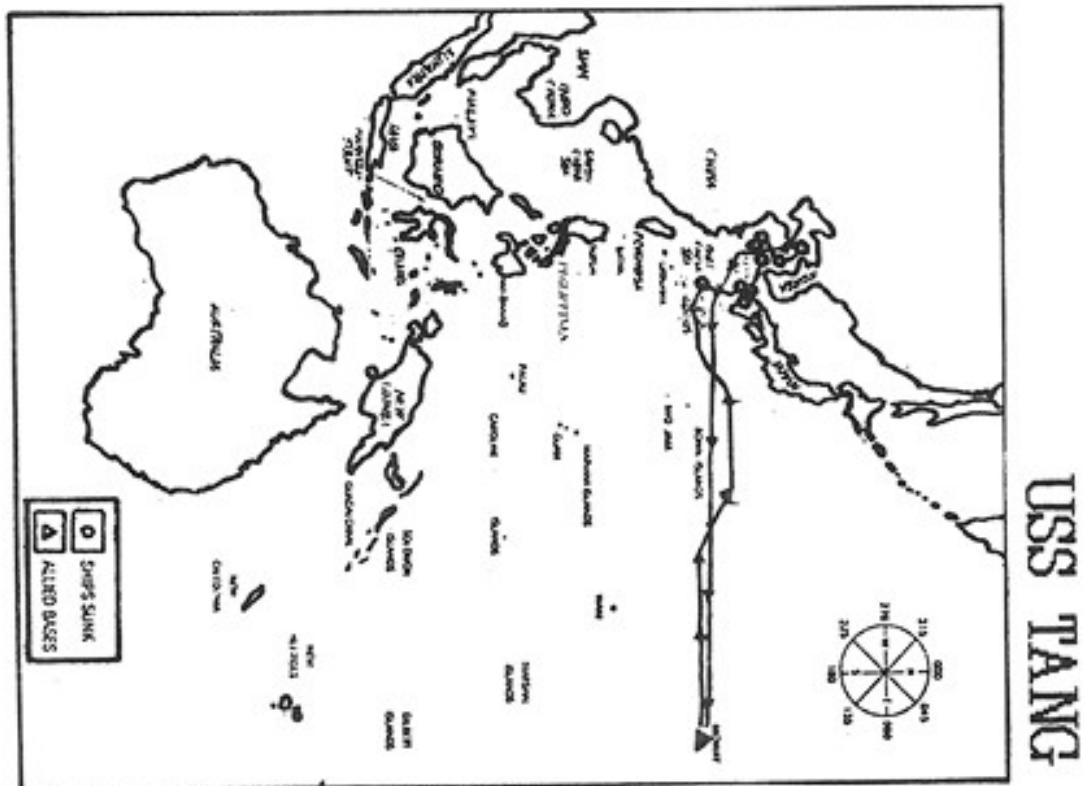
Pro zajištění přesnosti jsou všechny pohyby ponorky, dráhy torpéd, potápění a poklesy vypočítávány každé dvě sekundy simulovaného herního času. Někdy je vhodné některé simulace urychlit. Normální simulace je 4-krát rychlejší než ve skutečnosti. Opakováním stisknutím zrychlíš čas až na 32 násobek reálného času. Při stisknutí klávesy N se čas vrátí na normální hodnotu.

KONEC ÚKOLU, SKORE A ZARAZENÍ

Konvojové akce končí, když navolíš volbu konec hry, nebo úspěšně zlikviduješ celý konvoj. Úkoly válečného hlídkování (WAR PATROL) končí po návratu na jednu ze základen. Jiná možnost je potopení nebo najetí na mělčinu. Ve všech případech uvidíš scénu zobrazující všechny lodě, které jsi potopil a tvé konečné zařazení.

ZARAZENÍ: ENSIGN
 LIEUTENANT JC
 LIEUTENANT
 LIEUTENANT COMMANDER
 COMMANDER
 CAPTAIN
 VICE-ADMIRAL
 ADMIRAL FLEET ADMIRAL
 WGSC (nejlepší kapitán ponorky na světě)

Zmáčkní START pro další úkol. Pokud jsi byl úspěšný, budeš požádán o zapsání svého jména. Zadej své jméno a pak stiskni RETURN. Dvorana Slávy nahraje nejlepší zařazení a také skutečně potopené tonáže 5 ponorkami ve skutečné hlídce.



SCÉNÁŘE KONVOJOVÝCH AKCIÍ

Jsou to kratší scénáře, které tě zavedou do specifické historické situace. Jsou užitečné pro získání zkušeností, procvičení specifické taktiky, anebo máme-li málo času na WAR PATROL (válečné hlídkování).

1. PLUNGER ve vnitrozemním moři (Lt. Commander D.C.WHITE)

Den / pod vodou.

18.1.1942 - zeměpisná šířka 33-30 severně.
zeměpisná délka 135-38 východně.
Radar, parní torpéda, dostup 300 stop.

Americká ponorka PLUNGER hlídkující při jižním pobřeží Japonska zpozorovala nákladní loď s doprovodem, plující na východ velkou rychlosť. Tento scénář tě naučí torpédrovat pohyblivý cíl.

2. WAHOO proti konvoji (Lt. Commander MUSH MORTON)

Den / na hladině.

26.1.1943 - zeměpisná šířka 2-37 severně.
zeměpisná délka 139-42 východně.
Radar, parní torpéda, dostup 400 stop.

Americká ponorka WAHOO zpozorovala při pobřeží Nové Guiney malý japonský konvoj. Je to situace jako ve snu - konvoj bez doprovodu, včetně transportních lodí a velkých tankerů. Konvoj však vyslal radiovou depeši o pomoc a torpédoborce jsou na cestě.

3. HAMMERHEAD u Bornea (Commander J.C.MARTIN)

Noc / radar.

1.10.1944 - zeměpisná šířka 6-30 severně.
zeměpisná délka 116-11 východně.
Radar, parní torpéda, dostup 400 stop.

Když americká ponorka HAMMERHEAD hlídkovala u severního pobřeží Bornea, zachytily radar velký konvoj s doprovodem. Tvým cílem je jeden z hrstky japonských tankerů zbývajících v tomto stadiu války.

Tato scéna představuje noční boj proti strážnému konvoji. Postarej se, aby tě zpozoroval co nejpozději - použij přiměřenou rychlosť, minimální možný profil vzhledem k eskortě a zaútoč v době, kdy je eskorta na opačné straně konvoje.

4. SEARAVEN u TORGEL MLINGVI (Commander H.CASSEDY)

Objíždění.

13.1.1943 - zeměpisná šířka 9-12 severně.
zeměpisná délka 130-38 východně.
Radar, parní torpéda.

Někde mezi Filipíny a japonskou námořní základnou na TRUK LAGOON narazila americká ponorka SEARAVEN na konvoj. Jsi ve špatném postavení - vzadu za konvojem na denním slunci. Doporučuje se opatrné objetí konvoje, při čemž využij zrychlení času.

5. TAUTOB v noci (Lt. Commander SIEGLAFF)

Radar / noc / dobrá viditelnost.

15.3.1944 - zeměpisná šířka 42-25 severně.

zeměpisná délka 144-55 východně.

Radar, parní torpéda, zlepšený detonátor, dostup 400 stop.

U východního pobřeží Japonska se setkala americká ponorka TAUTOB s japonským konvojem. Noční útoky závisely mnoho na podmínkách viditelnosti. Při špatné viditelnosti se mohla nízká ponorka bezpečně přiblížit až těsně k cílům na hladině. Jestliže je viditelnost dobrá, vyžaduje to více opatrnosti.

6. GRAYBACK v Čínském moři (Lt. Commander J.R. MOORE)

21.10.1944 - zeměpisná šířka 26-48 severně.

zeměpisná délka 124-56 východně.

Radar, elektrická torpéda, dostup 400 stop.

Velmi obtížná situace. Tři radarem vybavené torpédoborce stráží konvoj.

Tvojí největší naději je periskopový útok za svítání, nebo za soumraku.

SCÉNÁŘE VÁLEČNEHO HLÍDKOVÁNÍ

Tyto scénáře prověří tvoji zručnost. Tvým úkolem je hlídat trasy japonských konvojů, najít, zaútočit a potopit maximum tonáže nepřátelského loďstva. Setkáš se s širokou paletou situací, přiležitostí a nebezpečí. Uvědom si, že každá ponorka je rozdílně vybavena (tvoje taktika by měla počítat se silnými a slabými stránkami tvé ponorky). Prohlédni si trasy japonských konvojů na mapě. Ve scénářích účinkují všechny tyto trasy pouze na začátku války. Jak válka pokračovala, plocha Pacifiku ovládaná Japonskem se zmenšovala tak, že se stalo obtížné najít konvoje v oblastech vzdálených od Japonska. Ve scénářích z pozdějších dob je proto nejsnadnější najít cíle ve vodách okolo japonských ostrovů.

1. USS TANG - MIDWAY PATROL, JUNE, 1944

(USS = United States Submarine, americká ponorka TANG /chaluha/, hlídka ze základny MIDWAY, červen 1944)

Radar, elektrická torpéda s vylepšenými detonátory, dostup 400 stop.

TANG byla druhá vedoucí ponorka s 24 ověřenými potopeními mezi 17.2. až 25.10.1944. Na trati hlídky pronikl její kapitán hluboko do Japonskem ovládaných vod Zlatého moře. Během pouhých 14 dnů potopila 10 nepřátelských nákladních lodí (včetně 4 za 1 den).

2. USS BOWFIN - FREEMANTLE PATROL, NOVEMBER, 1943
(americká ponorka BOWFIN /ohnutá ploutev/, hlídka ze základny FREEMANTLE, listopad 1943)

Radar, parní torpéda (se starými detonátory), dostup 400 stop

BOWFIN, se základnou v Austrálii, potopila 16 lodí pod vedením čtyř různých kapitánů. Při druhé hlídce plula z Austrálie přes Madagaskar strážit Filipíny. Po neplodném hlídkování u Filipín přeplula Jihočínské moře a dostala se do pobřežních vod Indočíny. Tady potkala dva konvoje a potopila pět lodí v průběhu tří dnů, navzdory mnoha problémům s torpédy.

3. USS GROWLER - SECOND PATROL, AUGUST, 1942
(druhá hlídka americké ponorky GROWLER /bručoun/ v srpnu 1942)

Hladinový radar, parní torpéda, dostup 300 stop.

Jedna z první flotily ponorek, které se zúčastnily boje, byla proslavena hrdinstvím svého kapitána H.W.Gilmorea. Po kolizi ponorky s japonským dělovým člunem Gilmore našel okamžitě se potopit, ačkoliv ležel těžce raněn na můstku, dávaje vlastní život za záchranu ponorky.

Druhá hlídka ponorky GROWLER začala v BRISBANE. U pobřeží FORMOSY potopila přes 15000 tun tonáže - vynikající hlídka v tomto kritickém stadiu války.

4. USS SEAWOLF - SEVENTH PATROL, OCTOBER, 1942 (sedmá hlídka americké ponorky SEAWOLF /mořský vlk/ v říjnu 1942)

Radar, parní torpéda, dostup 300 stop.

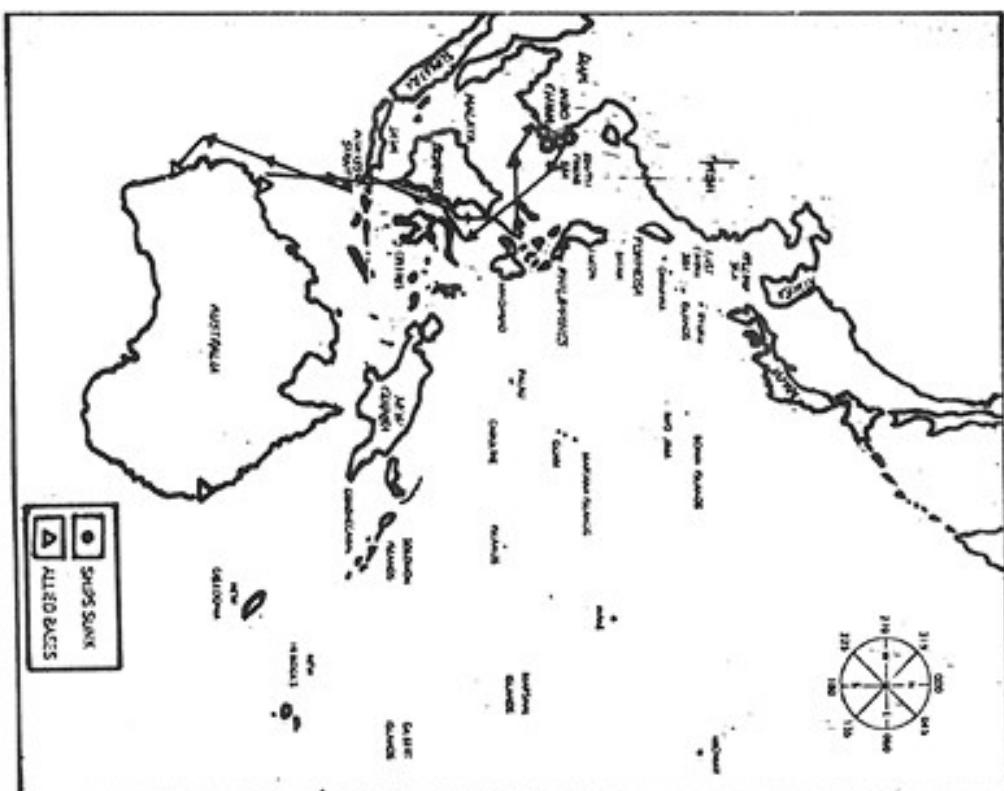
Jiná z prvních ponorek v Pacifiku, SEAWOLF, se stala jednou z nejúspěšnějších ponorek za války. Její druhá hlídka absolvovala pamětihonou bitvu proti japonským námořním silám u Vánočního ostrova.

5. USS SPADEFISH - SECOND PATROL, OCTOBER, 1944
(druhá hlídka americké ponorky SPADEFISH /ryba-rýč/ v říjnu 1944)

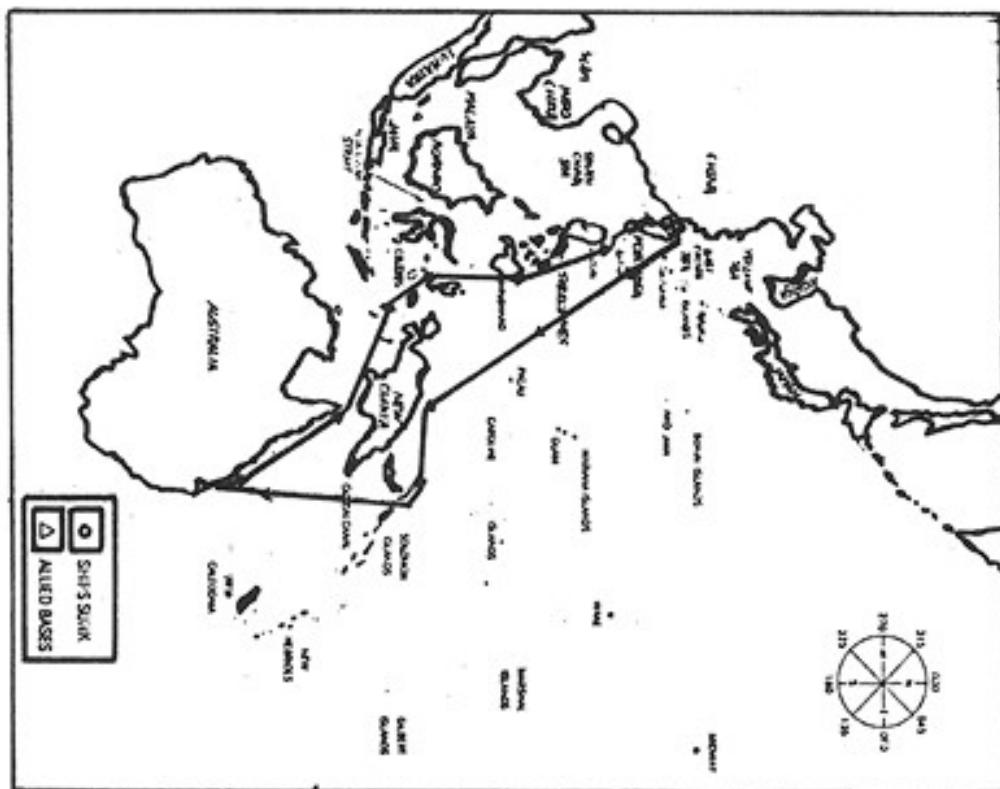
Radar, elektrická torpéda se zlepšenými detonátory, dostup 400 stop.

SPADEFISH vstoupila do akcí v roce 1944. V tomto stadiu války byla většina japonských eskort vybavena radarem. Navzdory jejímu pozdnímu začátku potopila 21 lodí s celkovou tonáží 88000 tun. Na své druhé hlídce, dva týdny po výjezdu z PEARL HARBORU, SPADEFISH narazila na těžce chráněný konvoj ve

USS BOWFIN



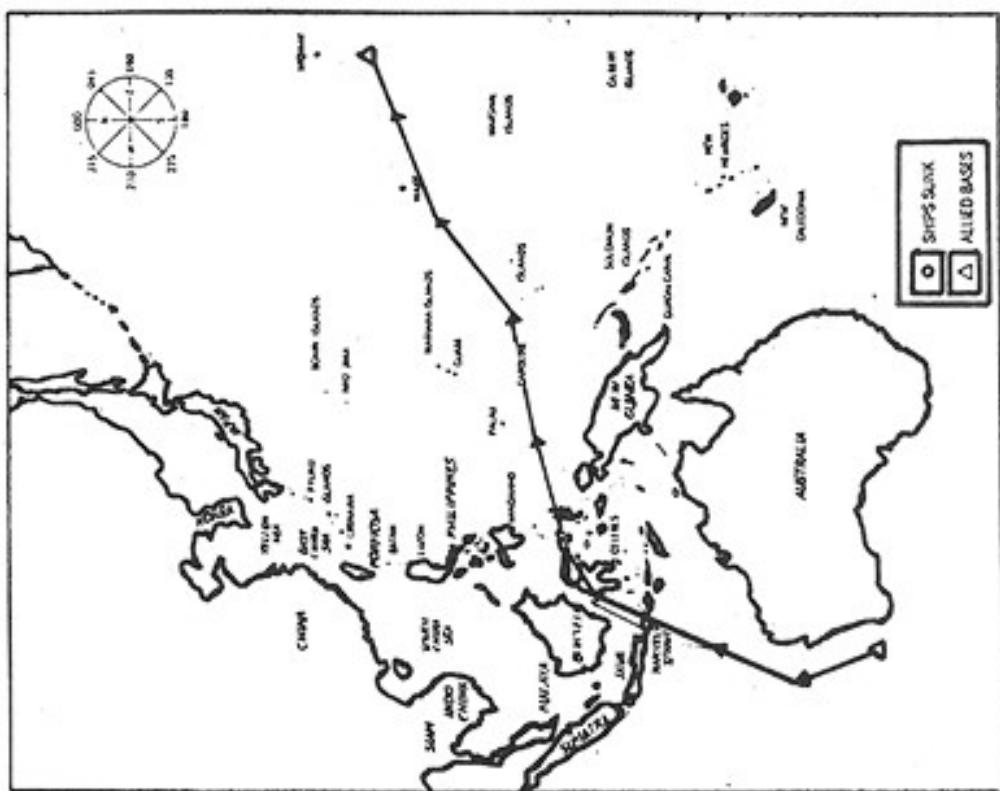
USS GROWLER



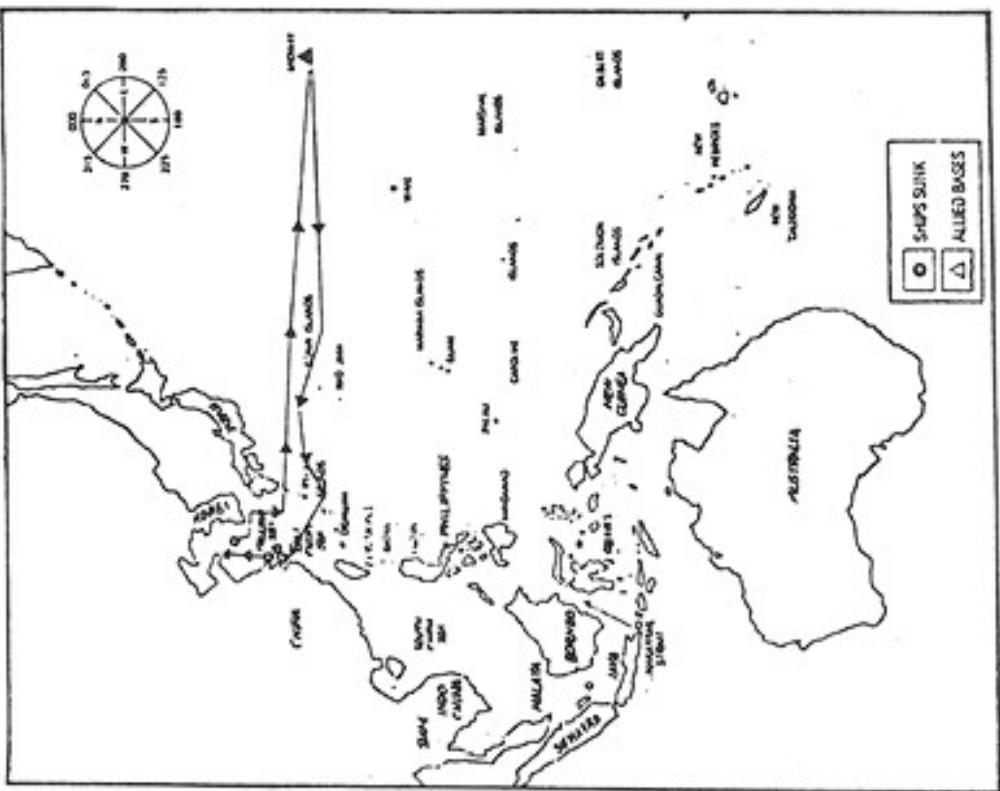
AZ Olomouc 7-8

- 50 -

USS SEAWOLF



USS SPADEFISH



Východočínském moři. Po vytrvalém sledování potopila srdce konvoje – dvacetitisícitonový doprovodný nosič JINYA.

PONORKOVÁ TAKTIKA

Úspěšný útok byl často výsledkem týmového úsilí celé posádky pod vedením kapitána. Obsluhy torpédu a strojníci se starali o údržbu torpédu a motorů. Zvukaři poslouchali nepřátelské lodě pomocí podvodních hydrofonů. Podle hluku lodních šroubů mohli odhadnout rychlosť nepřitele a natáčením hydrofonu i jeho polohu (BEARING). Radarová skupina sledovala nepřitele pomocí SJ hladinového radaru. Ve velitelské věži zakreslovala sledovací skupina polohu ponorky, nepřátelských cílů a doprovodných torpédoborců na útočné mapu. Identifikační skupina byla připravena k identifikaci typu nepřátelských lodí. Poté, když kapitán zavolal, co zpozoroval periskopem. Na můstku pozorovali pozorovatelé moře a pátrali po nepřátelských lodích. Jakmile se ponorka přiblížila k nepříteli, sledovací skupina zadávala do počítače torpédrových dat údaje o rychlosti, kursu, vzdálenosti a vizuálním kursu, nutné pro výpočet správného gyroskopického úhlu pro torpéda.

A na závěr této činnosti čekalo na kapitána kritické rozhodnutí, kdy jen malý rozdíl dělil úspěch od nezdaru. Pečlivě bral v úvahu počet doprovodných plavidel, typy lodí, viditelnost, hloubku, počet zbývajících torpédu, stav baterií, kurs, rychlosť konvoje a pak se rozhodl jak, kdy a kde zaútočit. Se svým nízkým profilem na hladině a možností se ponořit, nepozorovaně se přiblížit a překvapit nepřitele, musela ponorka počítat jako s životně důležitým prvkem při útoku.

Po vypátrání nepřitele vyžadoval úspěšný útok dobře promyšlené přiblížení o několik tisíc yardů, aniž by nepřítel ponorku zpozoroval, rychlé a rozhodné zamíření a odpálení torpédu, chytré využití rychlosťi, hloubky a teploty vody pro odvrácení nevyhnutelného protiútoku.

PŘIBLÍŽENÍ

Nejdůležitější při zpozorování nepřátelského konvoje bylo určit jeho kurs a složení. Potom padlo rozhodnutí o útoku. Dále mohl kapitán řídit ponorku do polohy před nebo za konvojem, zatímco stále zůstával nezjištěn nepřitem.

Během dne byl kapitán ponořen a čekal, než se konvoj přiblíží do palebného dosahu. V noci byl útok veden na hladině, ačkoliv se viditelnost měnila s mlhou a měsíčním svitem. Během soumraku a svítání byl použitelný periskop, ale přitom bylo obtížné vidět ponorku, což bylo ideální pro útok. Klíčem přiblížovací fáze bylo dosáhnout výhodného palebného postavení, aniž ponorku objevil nepřátelský doprovod. Následkem malé rychlosti pod vodou musela být většina manévrů vedena na hladině, což umožňovalo její objevení nepřitem.

Americký radar mohl zjistit lodě na vzdálenost více než 16000 yardů (8 mil). To dávalo ponorce počáteční výhodu, neboť japonští pozorovatelé ji mohli zpozorovat na vzdálenost 10000

yardů ve dne a 3000 yardů v noci. Pod vodou mohl sonar sledovat japonské lodě až na vzdálenost 6000 yardů, ačkoliv dosah se rychle zmenšoval, když byla ponorka v pohybu nebo v hloubce. Japonský sonar mohl zjistit rychle se pohybující ponořenou ponorku až na vzdálenost 5000 yardů, i když při maximální hloubce a nastavení na tichý chod bylo obtížné ponorku objevit.

Během přiblížení a úniku udržoval kapitán ponorku směrem k nepříteli (nebo od něj), aby mu poskytl minimální profil. Dokonce, i když byla ponorka pod vodou, poskytoval minimální profil nejmenší cíl pro sonary nepřátelských torpédoborců.

TABULKA VZDÁLENOSTÍ ZAMERENÍ PONORKY
(měřeno při rychlosti 10 uzlů)

podmínky	vzdálenost v yardech	
	den	
Na hladině:		
plný profil	20000	3000
minimální profil	8000	1000
V periskopické hl.:		
plný profil	6000	2000
minimální profil	2000	800
Ponořena:		
plný profil	2000	2000
minimální profil	800	800

Při ponoření ponorky pod vrstvu teplotního gradientu se možnost nalezení ještě snižovala.

TORPEDA

Základní výzbroj ponorky se skládala ze 6 předních torpédometů a 4 zadních. Celkový počet torpéd byl 24. 14 předních a 10 zadních. Znovu nabité torpédometu po výstřelu vyžadovalo asi 10 minut.

MARK 14

- parní torpédo s dosahem 4500 yardů při rychlosti 46 uzlů. Pro ochranu ponorky před předčasným výbuchem mohla být bojová hlavice ve funkci až po překonání 450 yardů. Torpédo MARK 14 bylo poháněno parou vyráběnou průchodem rozprášené vody přes hořák spalující alkohol. Zanechávalo za sebou na hladině bublinky, které směřovaly k ponorce. Torpédo bylo řízeno vnitřním gyroskopem. Toto složité zařízení trošku řadou závažných problémů. Mezi hlavní patřila tendence směřovat příliš hluboko, při čemž podplavalo cíl, a dále selhání roznětky MARK 6 při kontaktu s cílem. Oba tyto problémy byly v

průběhu války odstraněny.

MARK 18

- elektrické torpédo, které bylo zavedeno na konci roku 1944. Tato zbraň se pohybovala pomaleji než první torpédo. Její rychlosť byla 30 uzlů, ale nevystvářela proud bublinek jako její předchůdce. Velitelé ponorek již nebyli nuceni po první salvě torpéda rychle vrhat. Za ideálních podmínek se mohla potápět jedna loď za druhou, i když kolem zuřivě kroužila eskorta pátrající po neviditelném nepříteli.

Většina torpéda byla proto odpalována ze vzdálenosti 1000 až 3000 yardů. Nejlepší dráha torpéda byla kolmá ke kursu cíle. To poskytovalo největší možnou plochu cíle. Pro palbu ve směru přídě nebo zádě lodi byl zásah nepravděpodobný.

POČÍTAČ TORPEDOVÝCH DAT

Americké ponorky používaly počítač torpedových dat (TDC), který byl zpočátku analogový. Po zadání rychlosti, vzdálenosti a kursu cíle TDC počítal automaticky správnou dráhu pro torpéda. TDC vypočítal a zadal gyroskopický úhel přímo do gyrokopu, který řídil torpéda. Gyroskopický úhel spočítaný TDC vycházel z předpokladu stávající konstantní rychlosti a kursu.

Při očekávané změně kursu se proto často odpalovalo několik torpéda. Jedno torpédo mířilo nepatrne vpřed, jedno na a jedno nepatrne za cíl.

V této situaci se gyroskopický záměrný úhel automaticky přiříctá k vizuálnímu kursu v okamžiku odpalu. Například: máš nepřátelskou loď umístěnou přesně ve středu záměrného kříže, vizuální kurs je 090 stupňů (východ), kurs cíle je 180 (jih). TDC vypočítá gyroskopický úhel 10 stupňů. Když odpálíš torpédo, bude mít dráhu 100 stupňů (90+10) a mělo by zasáhnout cíl. Ve stejné situaci, pokud máš periskop v poloze 085 (nepatrne za cíl), tvé torpédo bude mít dráhu 095 stupňů (85+10). Toto torpédo by mělo minout cíl, ale může zasáhnout, když cíl bude kličkovat.

PALUBNÍ DĚLO

Většina amerických ponorek byla vybavena 4-palcovým palubním dělem. Toto dělo mělo dostřel až 8000 yardů a vysokou kadenci. Ačkoliv se používalo zřídká, bylo účinné v potápění těžce poškozených cílů. Dělo se také používalo jako poslední prostředek, když se ponorka musela vynořit, nebo měla příliš mnoho závad a nemohla se bezpečně ponofit. Dělo lze použít pouze na hladině. Použij záměrný kříž v periskopu pro zaměření. Dostřel je automaticky nastaven podle TDC. Použij klávesu < a > pro zvětšení a zmenšení dostřelu děla. Například: torpédoborec plující rychlostí 18 uzlů směrem k tobě ve vzdálenosti 400 yardů. Než ho zasáhne náboj z děla, urazí torpédoborec 200 yardů. Proto bys měl před výstřelem použít klávesu < pro

zmenšení dostřelu o 200-250 yardů. Při vzdálenosti 2000 yardů potřebuje náboj pouze polovinu času k dosažení cíle, takže bys měl dostřel snížit o 100-150 yardů. Současně může letět více nábojů. Když netrefíš cíl, uvidíš šplouchnutí vody. Při zásahu cíle uvidíš a uslyšíš výbuch. Dělo je vybaveno 80 náboji.

ONIK

Po zjištění ponorky nepřitelem se stal hlavním úkolem unik. Ponorka nebyla přizpůsobena pro souboj ve střelbě nebo najízdění ani proti jedinému torpédoborce. Obvyklou taktikou bylo co nejrychleji se ponorit a potichu plout.

Nepřátelská eskorta kroužila okolo poslední známé polohy ponorky a doufala, že zachytí odraz zvuku od jejího trupu. Udržování minimálního profilu a minimální hluk byly za těchto podmínek obzvláště důležité. Silný teplotní gradient mohl také poskytnout ochranu před nepřátelským sonarem. Unikání paliva nebo porucha strojů také usnadňovaly práci eskortě. Ponorky měly výhodu v malém poloměru otáčení a možnosti trvale sledovat hluk lodních šroubů eskorty.

Za extrémních podmínek mohla ponorka zkusit přesvědčit útočící torpédoborec o tom, že byla zničena, vypuštěním oleje a trosek, které vyplavaly na hladinu.

V praxi byla 20-uzlová rychlosť ponorky někdy dostatečná k tomu, aby ponorka ujela pronásledující eskortě.

JAPONSKE KONOVOJE

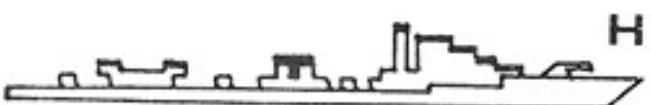
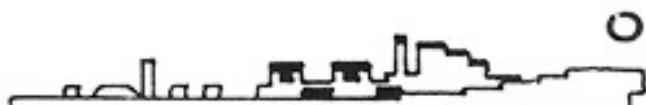
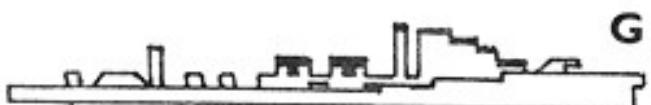
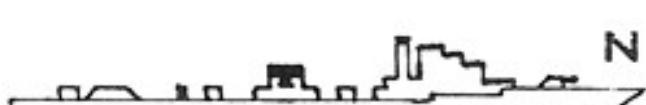
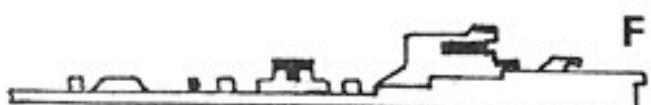
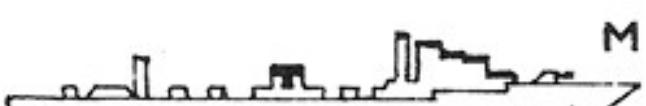
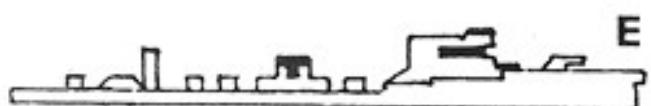
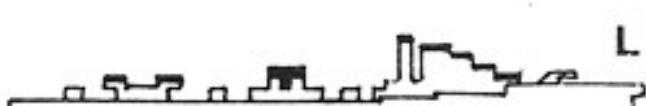
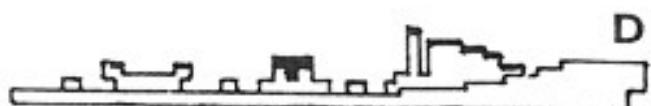
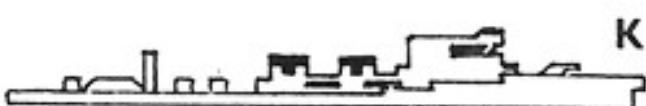
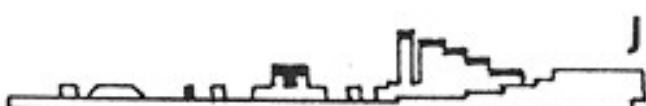
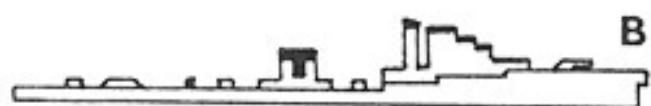
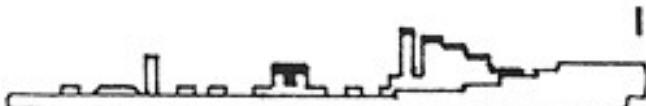
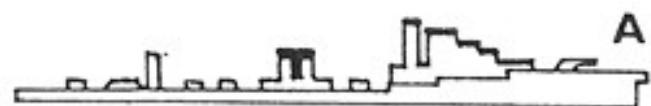
Japonské lodě se většinou pochybovaly v malých konvojích o 3 až 7 lodích. Občas mohly válečné a nákladní lodě cestovat osamoceně. S postupem války, kdy japonské ztráty rostly, se přidával stále větší počet doprovodných plavidel ke konvojům. Konvoje se skládaly z nákladních a dopravních lodí, tankerů a doprovodné eskorty. Mezi nejdůležitější cíle patřily tankery. Japonci byli kriticky závislí na příslušnu ropu, aby udrželi hlavní bitevní flotilu v provozu. Dopravní lodě byly také důležitým cílem. Tyto lodě přepravovaly výsadky při vpádech na vzdálené ostrovy. Pravděpodobně narazíš na tyto dopravní lodě na lodních cestách, které vedly přímo do Japonska. Nákladní lodě představovaly hlavní cíle japonského loďstva. Dopravovaly zásoby a vybavení.

Dopravná plavidla byla dvou typů:

1. TORPEDOBORCE – používaly se u zvlášt důležitých konvojů.
2. KAIROKANY – speciální lodě s protiponorkovými zbraněmi.

Oba druhy byly vyzbrojeny kanóny pro ničení ponorek na hladině, sonarem pro zjištění ponorek pod vodou a hlubinnými minami k jejich zničení. Ponorka mohla na hladině ujet KAIROKANU, který měl nejvyšší rychlosť 18 uzlů. TORPEDOBORCE mohly dosáhnout rychlosti až 30 uzlů.

Obratná a rychlá doprovodná plavidla s nízkým ponorem byla obtížným cílem pro torpéda, ačkoliv většinou stačil jediný zásah k jejich potopení.

SILENT SERVICE

JAPONSKÁ TAKTIKA

Japonské eskorty byly hrozným protivníkem. Jejich optické a sonarové vybavení bylo vynikající a japonské dělostřelectvo rovněž.

NÁČRTKY TAKTICKÝCH SITUACÍ

SITUACE 1 (Útok s objížděním)

Jsi v periskopické hloubce a právě jsi zpozoroval konvoj o rychlosti 10 uzlů, jehož bearing je 90 stupňů (východ). Základní kurs nepřitele jsi odhadl na 45 stupňů. Je právě poledne a zbývá 7 hodin denního světla. Konvoj je doprovázen alespoň jedním torpédoborcem. Tvoje torpédomety jsou plné a baterie plně nabita. Jaký je tvůj plán?

Je to obtížná situace, konvoj se pohybuje příliš rychle na přiblížení pod vodou. Opatrný kapitán by mohl nechat takový konvoj na pokoji a porozhlédnout se po snazší kořisti. Blázivě odvážný kapitán by mohl zkusit hladinový útok, ale vynořená ponorka není ve dne soupeřem pro torpédoborce. Zkušený kapitán by se pravděpodobně pokusil o taktiku objíždění konvoje.

Otoč se a pokračuj ponorem v plavbě od konvoje až na hranici viditelnosti (10000 yardů). Pak se vynoř a maximální rychlostí se dostaneš před konvoj, ale zůstaň z dosahu vizuálního pozorování. Sleduj konvoj na radaru. Jestliže eskorta opustí konvoj a pohybuje se směrem k tobě, byl jsi pravděpodobně zpozorován, a proto se okamžitě ponoř.

Provedení tohoto manévrů trvá nějakou dobu, použij proto zrychleného času pro urychlení simulace. Když se dostaneš před konvoj, ponoř se do periskopové hloubky a čekej, až dorazí konvoj. (Tato situace je podobná USS SEARAVEN.)

SITUACE 2 (noční hladinový přepad)

Hlídkuješ na hladině, když radar zachytí konvoj, jehož bearing je 45 stupňů. Je tmavá a nejasná noc. Radar ukazoval kurs nepřitele 180 stupňů a rychlosť 8 uzlů. Ochrannu konvoje zajišťují dva KAIROKANY. Co uděláš?

Je to vynikající situace. Jsi před konvojem a viditelnost je špatná. Základním požadavkem je, aby tě při přiblížení nezjistila eskorta. Použij rozumnou rychlosť a udržuj svoji příď směrem k eskortě. Tím budeš pro nepřitele pozorovatelný jen malým vizuálním úhlem. Měl bys dosáhnout ideálního palebného postavení ve vzdálenosti asi 1000 až 2000 yardů. Pokud se přiblížíš, když je eskorta na druhé straně konvoje, měl bys být schopen uniknout na hladině (KAIROKAN může plout rychlosťí pouze 18 uzlů). Hodně štěstí! (Situace je podobná scénáři USS HAMMERHEAD.)

SITUACE 3 (denní útok pod vodou)

Při pravidelném sledování periskopu jsi objevil konvoj plující směrem přímo k tobě ve vzdálenosti 4000 yardů. Dopravod je vpředu, následován 4 nákladními loděmi v kosočtvercovém uspořádání. Jednej rychle. Měl bys okamžitě odplout kolmo k trase konvoje a zaujmout boční palebné postavení. Protože budeš k nepříteli otočen bokem, měl by ses potopit, abys snížil možnost sonarového kontaktu. Počkej, dokud se dvě prostřední lodě nebudou překrývat. Torpéda, která minou bližší loď, pak mohou zasáhnout další loď.

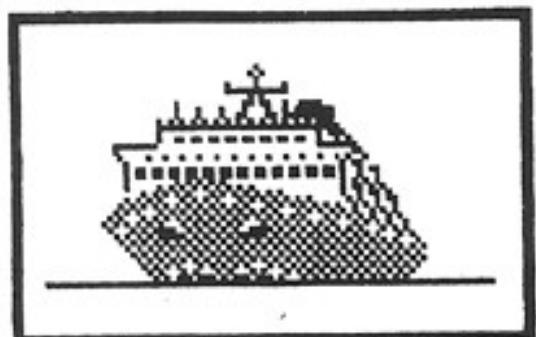
SITUACE 4 (únik před nepřátelskou eskortou)

Právě jsi vypustil 3 parní torpéda na obzvláště slibný tanker. Dva doprovodné torpédoborce zatím nezjistily tvoji přítomnost. Jsi v periskopové hloubce, je den. Je hrozně svůdné sledovat torpéda, jak se blíží k cíli. Udělal jsi to pouze jednou! Jakmile torpéda dosáhnou cíle, bublinky budou ukazovat přímo na tvé palebné postavení. Při rychlosti 26 uzlů tam bude torpédoborec rychle. Musíš okamžitě zmizet. Prchej před torpédoborem maximální rychlostí. Jestli bude torpédoborec blízko tebe, sniž rychlosť - snížení hluku. Dvě doprovodná plavidla mohou být velmi nebezpečná, neboť je obvykle nemožné udržet minimální sonarový profil k oběma lodím.

SITUACE 5 (únik v mělké vodě)

Máš potíže! Za tebou hoří tři nákladní lodě, které se staly cílem tvých torpéd. Ale hlavní eskorta jede přímo na tebe. Ale co je horší, hloubka vody je menší než 100 stop! Co teď?

Čeká tě dlouhé odpoledne. V této hloubce může být útok hlubinných min osudný. Nejlepší je využít malého poloměru otáčení ponorky, abys zabránil eskortě dostat se přímo nad tebe. Sleduj eskortu na útočné mapě a zkus předvídat její manévry. Použij maximální rychlosť. Tvou jedinou šancí je dostat se do hlubších vod.



POVELY Z KLÁVESNICE

šipky vlevo a vpravo - otáčení ponorky.
 (1. stisk - pomalu, 2. stisk - rychle)
 D - ponoření (DIVE)
 S - vynoření (SURFACING)
 RETURN - zastavuje vše uvedené funkce
 R - stroje vpřed/vzad
 P - periskop nahorá/dolů
 F - měřítko času
 N - normální časové měřítko
 G - střelba z děla
 I - indikace cíle (v periskopu)
 T - odpálení torpéda
 <,> - nastavení předstřelu děla
 A - zadávání úhlu předstřelu
 1 až 4 - rychlosť ponorky
 Ø - zastavení - tichý chod (SILENT SERVICE)
 W - zastavení programu (PAUSE MODE)
 SHIFT+E - vypuštění nouzových nádrží
 SHIFT+? - vypuštění trosek
 SHIFT+1 - mapy (STUDY MAPS)
 SHIFT+2 - můstek (GO TO BRIDGE)
 SHIFT+3 - pohled periskopem (USE SCOPE)
 SHIFT+4 - přístrojová deska (VIEW GAUGES)
 SHIFT+5 - hlášení o poškození (CHECK DAMAGE)
 SHIFT+6 - lodní deník (QUARTERMASTER'S LOG)
 SHIFT+8 - konec hry (u TORPEDO/GUN PRACTICE a CONVOY ACTION)
 - pokračování v hlídkování (u WAR PATROL)
 SPACE - návrat do kajuty

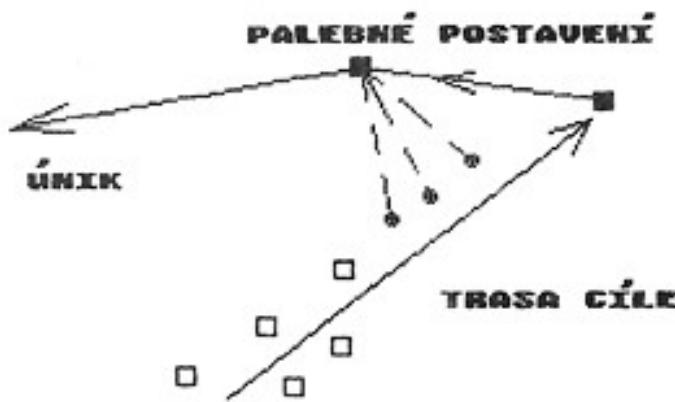
Příjemnou zábavu přeje
 Martin Kubečka



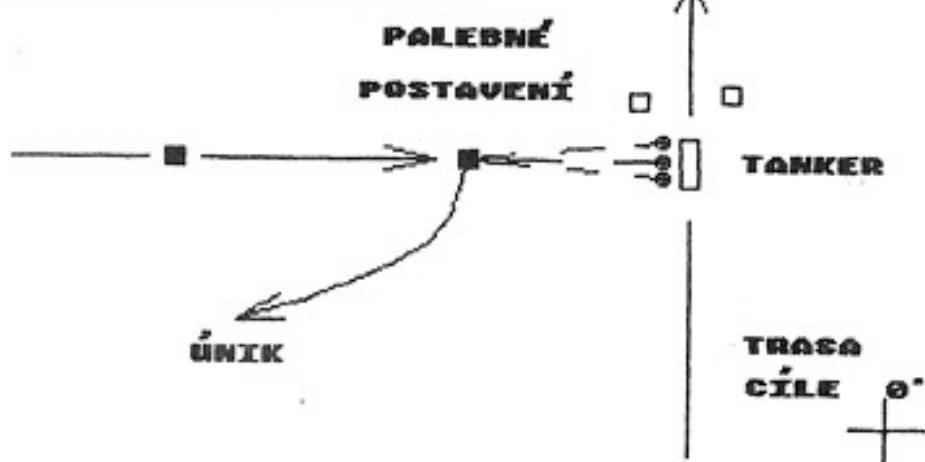
**SILENT SERVICE
SITUACE 2**

ÚNÍK**TRASA
CÍLE**

**SILENT SERVICE
SITUACE 3**

**TRASA CÍLE**

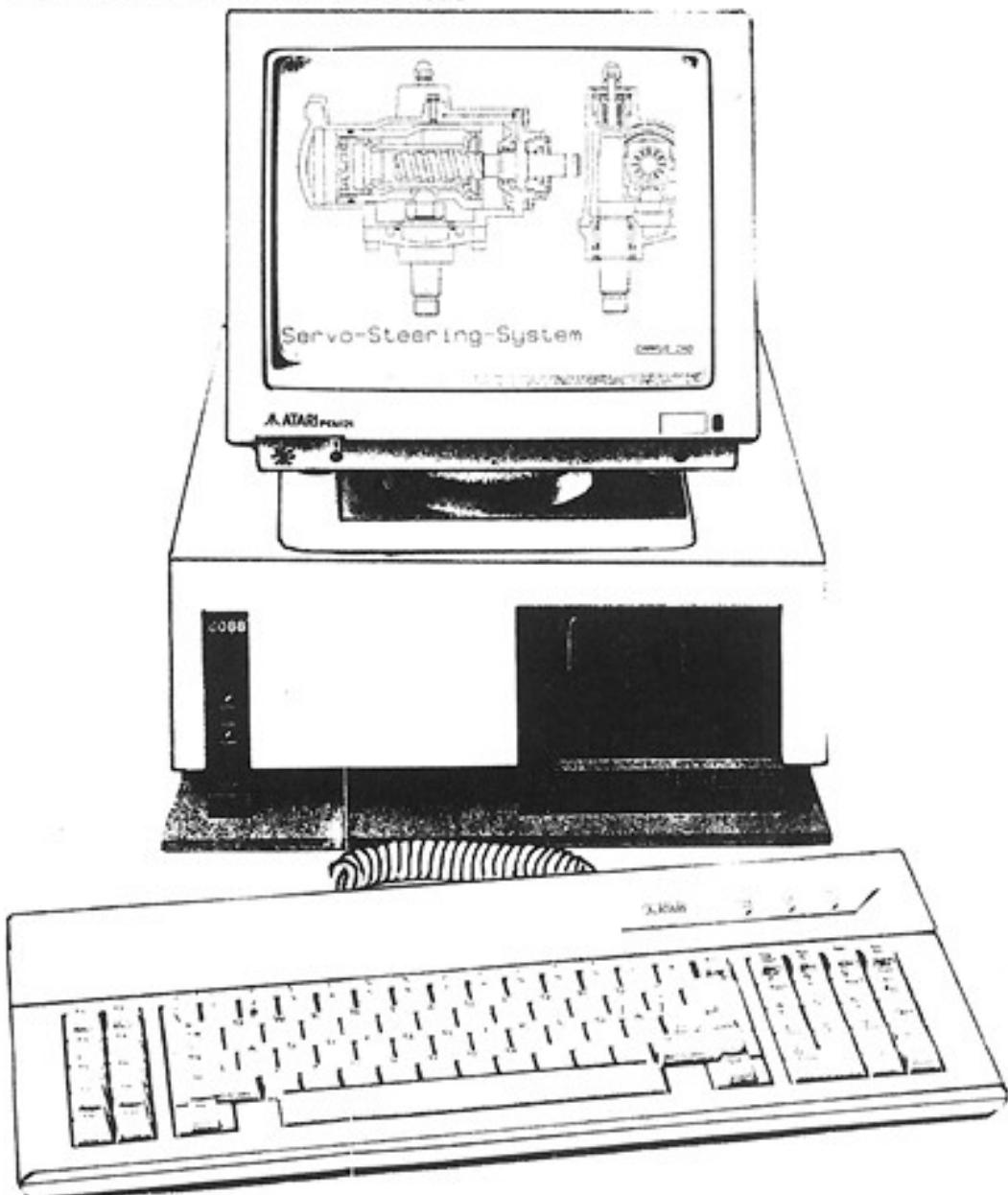
**SILENT SERVICE
SITUACE 4**

**TRASA
CÍLE**

Počítače z NSR

Vážení čtenáři, věřím, že někteří členové naší Atari rodiny začínají, stejně jako já, z různých důvodů pokukovat po výkonnějších a rychlejších počítačích. Důvodů je několik, například v práci mají nějaké to PC a potřebovali by si občas také něco připravit doma, či jednoduše vzrostly jejich osobní nároky na výpočetní techniku. Myslím si, že informace o Atari ST jsou mezi členy dostatečně rozšířeny a nákup ST náš problém stejně neřeší. Proto bych se chtěl podělit o informace, které se týkají možnosti nákupu počítače IBM PC kompatibilní. Nejdříve se o čistě opsané ceny z nějakého zahraničního časopisu, ale o informace získané osobně u jisté zásilkové organizace, která mi tyto exportní ceny nabídla (což znamená, že to jsou ceny s podmírkou vývozu zboží mimo jejich trh).

Tato organizace má bohaté zkušenosti se zasíláním počítačů i do socialistických států.



Exportní ceny:

Amstrad computer:

PC 1640 SD Mono	DM 1370 .-
PC 1640 SD EGA	DM 2045 .-
PC 1640 DD Mono	DM 1515 .-
PC 1640 DD EGA	DM 2270 .-
PC 1640 HD Mono	DM 1970 .-
PC 1640 HD EGA	DM 2725 .-

PC 2086 HD 12 MD	DM 2350 .-
PC 2086 HD 14 HRCD	DM 3180 .-
PC 2286 HD 12 MD	DM 3180 .-
PC 2286 HD 14 HRCD	DM 4100 .-
PC 2386 HD 12 MD	DM 5300 .-
PC 2386 HD 14 HRCD	DM 6250 .-

Apex computer XT:

FDisk Drive 5.25 / 360 kB , FD Controller, I/O card, keyboar

Model 1 10 MHz 512kB RAM	DM 980 .-
Model 2 10 MHz 512kB RAM < 1MB >	DM 1050 .-

Apex computer 286:

FDisk Drive 5.25 /1.2MB ,HD/FD controller .I/O card , keyboar

Model 1 12MHz , 1MB RAM < 4MB >	DM 1695 .-
Model 2 12MHz , 1MB RAM < 4MB >	DM 1895 .-
Model 3 12MHz , 1MB RAM < 1MB >	DM 1695 .-
Model 4 16MHz , 1MB RAM < 1MB >	DM 2100 .-
Model 5 12MHz , 1MB RAM < 4MB >	DM 1825 .-
Model 6 16MHz , 1MB RAM < 8MB >	DM 2245 .-

Apex computer 386 :

FDisk Drive 5.25 /1.2MB ,HD/FD controller .I/O card , keyboar

TOWER :

20MHz , 1MB DM 3325 .-

MTek computer 286 :

FDisk Drive 5.25 /1.2MB ,HD/FD controller .I/O card , keyboar

Desktop:	
10MHz, 512kB RAM, HD/FD contr., keyboard	DM 1781 .-
12MHz, 512kB RAM, HD/FD contr., keyboard	DM 1898 .-
16MHz, 1MB RAM, HD/FD contr., keyboard	DM 2358 .-
 MTek computer 386SX :	
FDisk Drive 5.25 /1.2MB ,HD/FD controller ,I/O card , keyboar	
 Desktop:	
16MHz, 1MB RAM,HD/FD contr.,keyboard	DM 2933 .-
 MTek computer 386 :	
FDisk Drive 5.25 /1.2MB ,HD/FD controller ,I/O card , keyboar	
 Desktop:	
16MHz, 1MB RAM,HD/FD contr.,keyboard	DM 3450 .-
20MHz, 1MB RAM,HD/FD contr.,keyboard	DM 3968 .-
 Tower:	
16MHz, 2MB RAM,HD/FD contr.,keyboard	DM 4658 .-
20MHz, 2MB RAM,HD/FD contr.,keyboard	DM 5693 .-
 Hard disk drive:	
Seagate, ST 225, 20MB,MFM	DM 486 .-
Seagate, ST 251-0, 40MB,MFM	DM 748 .-
Seagate, ST 251-1, 40MB,MFM	DM 796 .-
Seagate, ST 4096, 80MB,MFM	DM 1415 .-
Seagate, ST 157N, 49MB,SCSI	DM 830 .-
Seagate, ST 296N, 80MB,SCSI	DM 1170 .-
NEC D 3142 , 44MB , MFM	DM 911 .-
NEC D 3661 ,118MB , ESDI	DM 1872 .-
NEC D 5655 ,140MB , ESDI	DM 2248 .-
NEC D 5662 ,319MB , ESDI	DM 4373 .-
 Controller:	
XT controller HD	DM 120 .-
AT controller HD	DM 218 .-
WD 1007A controller HD-FD ESDI	DM 570 .-
SCSI Host adapter 01	DM 58 .-
SCSI Host adapter 02	DM 98 .-
 Streamer:	
Colorado Jumbo 40MB	DM 808 .-
Colorado AB-10, XT/AT controller	DM 240 .-
DC 2000 Cassette	DM 70 .-

EGA Option (adapter a monitor):

EGA Option 1024 * 768	DM 1436 .-
EGA Multiscan Option	DM 1560 .-

Grafický adapter VGA:

VGA Mira plus	DM 386 .-
VGA 2Max	DM 462 .-
VGA 3Max	DM 748 .-
Video Seven Vega VGA	DM 765 .-
Video Seven FastWrite VGA D-RAM	DM 984 .-
Video Seven FastWrite VGA V-RAM	DM 1445 .-

Monitory:

TTL (Hercules komet.) Monochrom	DM 250 .-
VGA Mono INTRA	DM 286 .-
VGA Mono TVM	DM 374 .-
VGA Color INTRA	DM 823 .-
VGA Color TVM	DM 1062 .-
Multiscan Color INTRA	DM 1065 .-
Multiscan Color TVM	DM 1249 .-

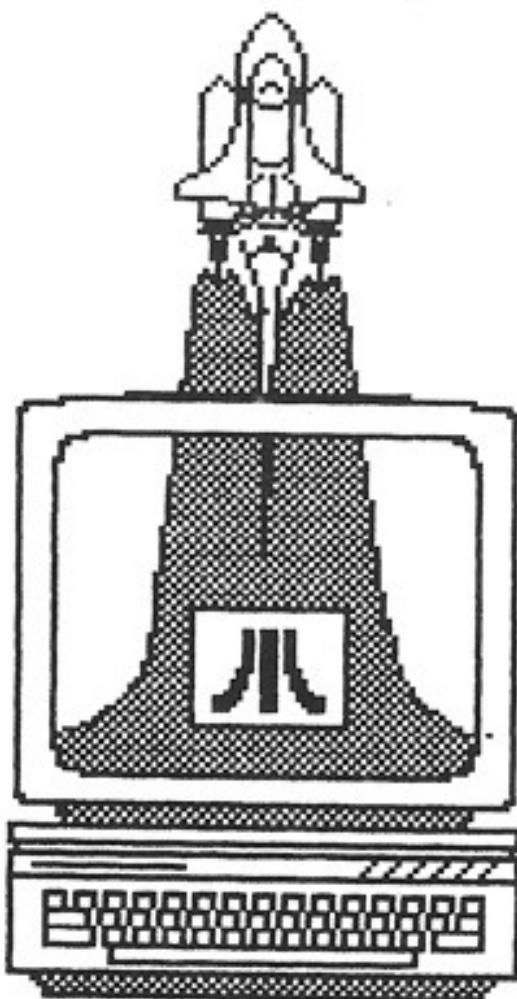
Tiskárny:

Amstrad DMP 3160 (9 pin A4)	DM 455 .-
Amstrad DMP 4000 (9 PIN A3)	DM 645 .-
Amstrad LQ 3500 (24 pin A4)	DM 645 .-
Amstrad LQ 5000 (24 pin A3)	DM 1058 .-
Star LC 10 (9 pin A4)	DM 398 .-
Star LC 20 (24 pin A4)	DM 680 .-
Epson LX 400 (9 pin A4)	DM 480 .-
Epson LQ 400 (24 pin A4)	DM 650 .-
Epson LQ 850 (24 pin A4)	DM 1350 .-
Epson LQ 1050 (24 pin A3)	DM 1780 .-

V přehledu byly použity běžně používané značky a zkratky, takže by se měl zorientovat i začátečník. Chtěl bych upozornit, že ceny jsou aktuální k listopadu 89. Vzhledem k dodacím lhůtám tiskárny uvádím toto datum pro případné změny. Pokud budete mít vážný zájem a budete potřebovat další bližší informace, mohu se pokusit vám je zprostředkovat.

Adresa : ins. Podmolík Leopold
Urxova 4
772 00 Olomouc

ÚSPĚŠNÝ START DO NOVÉHO ROKU 1990



ZO SWAZARAH
ATARI IKLADNEC
OLOMOUC

Obsah Atari zpravodaje Olomouc 7-8/1989:

Název:	str:
1. Atari klub ST	1
2. Od Kyan PASCALu k Turbo PASCALu!	2
3. Třídění tabulky čísel	12
4. PMWindows	18
5. BASICové melodie	20
6. Animace na počítači ATARI 130 XE	28
7. MiSe - program HATABS.BAS	31
8. MiSe - program C256.BAS	31
9. Silent Service - manuál hry	32
10. Počítače z NSR	60
11. PF 1990	64

Zpravodaj AK Olomouc 7/8 1989
NEPRODEJNÉ, odběr vázán na příspěvek.

Odpovědný redaktor: ing. Kopečný Pavel
Odborný redaktor : Hrdlička Jiří

Neprošlo jazykovou úpravou.
Předáno do tisku: XI.-1989

Tisk povolen OK ONV Olomouc, 0380500387
Tisk: MTZ 11 Olomouc