



19 settembre 2015

VOLUME 1, N°9

Questa newsletter italiana nasce da un accordo avuto con Chris Schneider che ha acconsentito alla traduzione italiana della sua *Newsletter Shift838*. Vuole essere una nuova risorsa per i computer **TI-99/4A** e **GENEVE 9640**, in modo da divulgare e mantenere aggiornati tutti gli utenti che ancora oggi si interessano a questi computer.

In questa edizione ci siamo dedicati a **Danny Lousberg** altrimenti noto come **'The Mole'** (*La Talpa*), che sta sviluppando un nuovo ed avvincente gameport del gioco **'Alex Kidd'** sul TI-99/4A. Leggete ! sono sicuro che vi piacerà ☺

Per approfondire questo argomento potete accedere al seguente link di AtariAge:

<http://atariage.com/forums/topic/226068-alex-kidd-port/>

Altri argomenti trattati in questo numero:

- **Geneve 9640 Menu System** adattato per il TI-99/4A;
- **Concorso TI Gaming by Owen Brand**, questo mese si gioca a: **Miner 2049er!**;
- **Lezioni di Assembly 1 – by Lee Stewart** (in Inglese);

Buona Lettura !

[Ciro Barile](#)

Chris intervista:

The Mole

Chris: Da quanto tempo ti trovi nel mondo TI ?

The Mole: mio padre acquistò un TI quando avevo 4 anni, nel 1982. Mi ricordo molto bene di aver giocato a TI Invaders per la prima volta per una mezz'ora o giù di lì prima di andare a letto. Fu molto eccitante e non riuscii a dormire, così alla mattina presto scesi furtivamente al piano di sotto e trovai i miei genitori a giocare e a cercare di fare il punteggio più alto. Avevano giocato tutta la notte. Pochi anni più tardi, quando avevo 10 anni, andammo in un appartamento al mare (dove viveva la famiglia di mia madre) e lì trascorremmo la maggior parte delle nostre vacanze. Ma chi conosce come solitamente è il tempo atmosferico in Belgio, sa bene che per la maggior parte dell'anno si finisce per rimanere bloccati in casa. Ho giocato tantissimo con il TI, ore ed ore di partite, e quando finalmente mi stufai di Parsec, Carwars, Invaders e di altri giochi in Extended BASIC, iniziai a pasticciare con i listati di alcuni giochi trovati su una rivista da digitare. Come sai, è noioso mettersi a copiare quelle lunghe liste, modificare la grafica qua e là, aggiungere vite o accelerare i proiettili, aggiungere il mio nome alla schermata del titolo. Questo finché mio padre mi diede il Reference Manual dell'Extended BASIC e mi insegnò come creare i miei primi programmi BASIC.

Chris: Che cosa in particolare ti spinse verso il TI-99/4A ?

The Mole: diciamo che fondamentalmente è stato il computer che abbiamo tenuto e usato per un lungo periodo di tempo. Ero piuttosto geloso dei miei amici che possedevano il C64, la qualità dei giochi su quel sistema era un bel po' avanti rispetto al TI. Prendemmo un Sega Master System e alla fine un Genesis, così mentre continuavo a programmare sul TI, potei soddisfare la mia voglia di giocare su sistemi più potenti.

Chris: Puoi illustrarci qualche altro programma o progetto che hai realizzato?

The Mole: non ho molto da raccontare in realtà, sono ritornato sul TI-99 solo un paio di anni fa, quando ho scoprii l'emulatore **TI-99/SIM** (di M. Rousseau) e la **GCC** port (*GNU Compiler Collection* che è un compilatore per il linguaggio C ndt) creato da Insomnia (sono stato un utente Linux per lungo tempo e l'ho usato per lavorare come sviluppatore C, dove realmente risiedono le mie capacità e i miei interessi). Il mio primo progetto pubblico fu probabilmente il mio tentativo di fare un raytracer

per il TI, di cui ancora oggi è possibile trovare alcuni miei video su YouTube. Alla fine ho cercato di crearci qualche cosa, ma era troppo lento per un action game, così lasciai perdere. Durante le sessioni di codifica di Alex Kidd ho usato però quel codice per testare il mio ambiente di sviluppo della GPU (Graphic Processor Unit) della F18A. Ripresi quel programma esclusivamente per fare un gioco usando la F18A, chissà ... Ho anche rilasciato il codice sorgente per lo strumento PC-Side (una vecchia versione) che crea i dati grafici per lo scrolling in Alex Kidd. Oh, e ho fatto anche la grafica per schermata iniziale nelle versioni più recenti di Magellan.

Chris: puoi spiegare agli utenti quale sia la caratteristica preferita attualmente del TI che ritieni "TUA"?

The Mole: è difficile rispondere, non c'è niente che mi esalta in modo particolare. E' tutto l'insieme del pacchetto completo e una buona dose di nostalgia che mi fa amare il sistema. Mi diverte molto e se mi stai chiedendo che cosa mi piace di più del TI, probabilmente dovrei elencarti troppo cose ...anche se è il mio sistema preferito in assoluto, è difficile per me individuare cosa mi piace particolarmente più di altre.

Chris: dove metti in pratica le tue abilità, usando qualche emulatore del TI o sull'hardware reale e perché?

The Mole: beh, io uso un cross-compiler, così faccio tutto sul mio Mac. Io ho un bel po' di roba davanti a me (controllate la foto allegata della mia scrivania incasinata) in questo modo posso spesso controllare il mio lavoro e lo sviluppo risulta molto più facile usando l'emulatore. Non potrei fare molti progressi usando direttamente il TI, io non sono una di quelle persone che utilizza tutti i giorni il TI come un computer di lavoro.

Chris: perché hai scelto di fare un porting di Alex Kidd per il TI-99/4A?

The Mole: Da bambino, quando i miei amici avevano il Nintendo, io possedevo un Master System. Il gioco che accompagnò la console fu *Alex Kidd in Miracle World* ed ho assolutamente amato quel gioco. Aveva una profondità di colore e una giocabilità che mi inducevano a rigiocarlo sempre e anche oggi ancora gioco alcuni livelli. Ho sempre desiderato poter creare un gioco del tipo platform a scorrimento orizzontale sul TI, proprio perché il nostro computer ha una carenza grave in questo stile di giochi che invece era molto popolare in tra le console per videogiochi e per home computer che erano sul mercato oltre il TI. Infine, ho pensato che sarebbe stato meglio se il mio primo gioco l'avessi impostato usando un gioco esistente, invece di dover partire da zero.

Chris: A quanto ho capito, il tuo gioco è completamente codificato in linguaggio C. Secondo te, avendo dato dimostrazione della capacità di questo linguaggio, altri programmatori, potrebbero prenderlo in considerazione per programmare giochi futuri sul TI-99 ?

The Mole: tutta la logica del gioco è fatta in C, con alcuni piccoli frammenti di assembly qua e là per comunicare con l'hardware. Io in realtà ho trascorso un paio d'ore per sostituire la maggior parte del codice assembly con il codice C, per motivi di gestione. Credo davvero che il port di GCC di Insomnia genera un codice simile all'assembly in termini di velocità, ma i programmatori più esperti di assembly lo ritengono un ambiente di sviluppo ideale per i giochi d'azione. Mi auguro che possa ispirare altri programmatori che forse sono un po' scoraggiati dall'assembler e pertanto potrebbero provare a codificare qualcosa in C. Proprio come il BASIC, il 'C' è un linguaggio di semplice comprensione e molto strutturato (facendo attenzione e seguendo alcune semplici regole). È infinitamente più veloce del TI BASIC, quindi potrebbe essere una scelta eccellente per alcune delle persone che stanno cercando di trovare un linguaggio di programmazione migliore. E per coloro che sono nuovi per il C, consiglio di utilizzare la bella libreria "libti99" di Tursi, che fornisce l'equivalente di un sacco di CALL del BASIC (non funzionano però in modo simile e quindi avrete bisogno di imparare come usarle) in modo da non doversi preoccupare del problema di comunicazione diretta con l'hardware.

Chris: con il porting di Alex Kidd, ti sei attenuto al gameplay originale oppure hai cambiato alcuni aspetti a tuo piacimento, mantenendo l'ambiente del gioco originale?

The Mole: ho trascorso un po' di tempo cercando di mantenere l'ambiente più vicino possibile all'originale. Penso che questo sia ciò che a fatica trovo nei giochi più antichi, come ad esempio Miner2049'er, Jungle Hunt o anche Donkey Kong: il personaggio semplicemente non si controlla molto bene, quando fa salti ad arco e accelerazioni/decelerazioni veloci. D'altra parte, ho dovuto trovare un gran numero di punti di incontro per poter far funzionare il gioco sul TI, in particolare per alcune funzionalità e nel disegnare i livelli.

Chris: Se hai cambiato qualcosa, puoi dire con poche parole cosa hai fatto e perché?

The Mole: c'è un gran numero di cose che, per motivi tecnici, sono differenti:

- Per esempio, non c'è spazio sufficiente per i pattern nel fare una distinzioni tra sacchi di denaro grandi e sacchi piccoli o per aggiungere casse speciali che si

trovano in alcuni livelli. Così ho definito una sola dimensione del sacco e ho nascosto gli oggetti speciali nei normali contenitori a 'stella'.

- Il gioco originale ha sia dei livelli di scrolling verticali sia livelli di scrolling orizzontali. Avrei bisogno di potenziare un po' il mio motore grafico per gestirlo sul TI ma preferirei arrivare ad un risultato prima di eccedere con caratteristiche particolari.
- Non sono sicuro di includere partite di morra cinese. Non ho mai sentito che giocare partite di morra cinese ("sasso-carta-forbici") fosse un buon esempio di gioco d'azione. Non ho ancora deciso in merito.
- Lo shop che si trova nel gioco originale non farà parte dei livelli, il quale, invece sempre per motivi tecnici, verrà presentato prima di iniziare un livello, dal momento che lo shop potrebbe escludere troppi pattern e limiterebbe troppo i colori necessari all'algoritmo di scrolling.
- Nel gioco originale c'è una storia, che viene raccontata da personaggi che si incontrano alla fine di certi livelli. Ciò sarà sostituito da schermate della storia in via separata, sempre a causa dell'insufficienza di pattern.

Dato che mi sono visto costretto a fare questa serie di modifiche, ho deciso di non fare una port tradizionale, ma di creare un nuovo gioco nello stesso universo, usando la maggior parte dello stesso gioco, ma con livelli nuovi e originali. I livelli attuali nella demo sono copiati dal gioco originale, ma nella versione finale ci saranno per lo più livelli originali.

Chris: quali sono alcuni dei principali ostacoli che hai dovuto superare per questo porting ?

The Mole: il gioco ha un ambiente relativamente dinamico tale da consumare un sacco di RAM. In origine, il gioco doveva nascere per essere eseguito da disco ma ne esaurii la capienza. Ho speso un sacco di tempo per elaborare metodi di compressione che avrebbero dovuto avere la caratteristica di decomprimere/aggiornare/ricomprimere la mappa dati in tempo reale e velocemente, o permettere la manipolazione dei dati nella mappa, senza decompressione/compressione, ma tutto si è rivelò essere troppo lento per far girare il gioco a 60 fps. Sono sicuro che un programmatore più talentuoso di me sarebbe in grado di scoprirlo, ma per come stavano andando le cose, la faccenda mi ha portato quasi al punto di abbandonare (o almeno notevolmente semplificare) il progetto. Lo lasciai fermo a quel punto di sviluppo per ben 8 mesi circa, prima di

concludere che l'approccio giusto era il rilascio su modulo. Sarà ancora necessaria l'espansione di memoria 32k, ma tutto il codice viene eseguito da modulo lasciando la maggior parte di quei 32k disponibile per i dati del gioco.

Chris: so che ci sono stati nuovi giochi specificamente progettati per utilizzare la nuova scheda video F18A per migliorare la grafica e non gireranno sul TI-99/4A standard se non aggiungendola. Il tuo lavoro sul porting di Alex Kidd richiederà un TI-99/4A standard o sarà necessaria la grafica avanzata della F18A?

The Mole: il gioco potrà essere eseguito su un TI standard con espansione di memoria 32K, ma gli utenti che hanno un F18A installata vedranno notevolmente migliorata la grafica. Il gioco rileverà automaticamente che tipo di VDP è presente e sceglierà di conseguenza il percorso di rendering migliore. Ciò significa che ci verrà rilasciata una versione per tutti i sistemi. Rasmus ha fatto alcune cose meravigliose sulla F18A e che mi ha davvero ispirato a considerarlo.

Chris: Guardando la schermata della mappa, è molto diversa da quella originale (credo personalmente sia migliore e meno ingombrante), ma perché ti sei orientato a farla in questo modo?

The Mole: ho sempre pensato che le mappe più stilizzate come si potevano vedere in Mario o in Zelda si adattavano meglio alle capacità grafiche di presenti in quegli anni nei computer/console. La maggior parte della grafica originale di Alex Kidd era molto in stile cartoon e stilizzata, fatta eccezione per la schermata della mappa. Non ho mai capito perché lo fecero, ma a me sarebbe piaciuto provare a progettare la mia mappa partendo da zero. Fare grafica è uno dei modi per fare una pausa dalla programmazione ma senza allontanarsi dal lavoro sul progetto. E' un modo per chiarirmi la mente quando sto affrontando un brutto bug o un pezzo particolarmente impegnativo su nuove funzionalità. Mi sono molto divertito facendo di recente gli sfondi per la versione con l'F18A.

Chris: quanti miglioramenti pensi siano stati apportati al tuo gioco da quando ti sei confrontato con i feedback degli utenti sul forum AtariAge?

The Mole: è difficile da dire. Sono fortunato, nel senso che ho potuto appoggiarmi a giganti come Rasmus, Tursi e molti altri per alcuni argomenti tecnici tra i più difficili. Un esempio per il quale ho avuto un ampio riscontro di feedback veramente utili è stato in merito al progettazione della mappa, in cui più persone hanno effettivamente migliorato e contribuito con elementi che poi sono finiti nel progetto finale. Ho sempre pensato che questa sia una grande vetrina che mostra come la nostra piccola comunità può lavorare bene insieme.

Posso dirvi una cosa, però, che se non fosse per le molte reazioni positive per il gioco, probabilmente non avrei portato avanti il progetto per così tanto tempo. In genere, nei miei progetti, tendo ad arrivare alla fase di concept e poi perdo interesse. Il continuo sostegno e le gentili parole da parte di tutta la community mi hanno veramente aiutato a motivarmi maggiormente.

Chris: la musica è sorprendente, è stata codificata da te ed elaborata per tentativi ed errori o è stata importata dal gioco originale semplicemente ?

The Mole: tutto il merito per la musica e gli effetti sonori va alla stupefacente libreria musicale SPF di Tursi. Non credo che ci sia tanta gente che possa capire come ha impattato il lavoro di Tursi , ma una buona parte delle nuove demo e dei nuovi giochi che hai visto di recente fanno uso dei suoi eccellenti strumenti. Mi è stato permesso di scaricare file musicali principalmente dai giochi originali in formato VGM da www.smspower.org e di convertirli in un formato compresso che è direttamente utilizzabile da C e di assemblerli usando la libreria di playback altamente ottimizzata da Tursi.

Chris: puoi spiegare brevemente perché questo gioco è attualmente diventato un modulo SSS da 512K per chi utilizza il computer reale ?

The Mole: come ho detto precedentemente, ho bisogno di un sacco di RAM per memorizzare i dati per ogni livello mentre il gioco è in esecuzione, con un volume dei dati in funzione della lunghezza del livello. In media occorrono 3K per la musica, 12K per la NameTable e le collisioni e un paio di K per altre cose varie. Ho anche bisogno di circa 24K per il codice, cosicché quella da 32K non basta. Ho pensato di usare una SuperCart, ma questa mi avrebbe aggiunto solo 8K in più, che sarebbe anche stata sufficiente, ma sempre ai limiti con lo spazio. Un'altra opzione era di usare memorie (S)AMS, ma non tutti ne sono in possesso ed io volevo raggiungere più utenti possibili. Così ho deciso di eseguire il gioco da modulo e evitare l'uso dei dischi. La ragione per cui mi sono rivolto alle schedine da 512K (e non qualcosa di più piccolo) è perché necessito di un po' di spazio di archiviazione per i dati dei livelli dei due diversi chip VDP. Nel progetto originale basato su disco, avevo programmato di fare due versioni scaricabili separatamente, ma per il rilascio su modulo si deve essere sicuri di avere una sola SKU per tutte le versioni.

Chris: produrrà i moduli e li venderà con i manuali, ecc. o hai intenzione di fornire solo il codice per l'utente che poi li può masterizzare su una EPROM sulla schedina del modulo oppure pensi ad entrambe le soluzioni?

The Mole: attualmente sto progettando di rilasciare moduli fisici in scatolati con i

manuali e le opere prime e rilasciare l'immagine finale della modulo per coloro che si vogliono programmare una EPROM quando avrò sufficienti pre-ordini per coprire i costi del setup. Spero davvero che io possa incontrare quella qualità che si dovrebbe percepire per esempio quando *CollectorVision* rilascia un suo prodotto (vedi p.e. Mr Chin a: <http://www.collectorvision.com/TI99/ti99-mrchin.htm> ndt).

Chris: so che il gioco non è completamente finito, ma se dovessi stimare quante ore sono state investite in questo progetto, quanto tempo ancora pensi che occorrerà per portarlo a conclusione?

The Mole: ci ho lavorato a intervalli e per brevi periodi di tempo nel corso degli ultimi due anni o giù di lì. E' difficile stimare il numero di ore, ma certamente parecchie settimane, se non mesi di programmazione a tempo pieno. Con uno sforzo d'immaginazione non mi considero uno cattivo programmatore, ma quando vedo la velocità con cui persone come Sometimes99er o Rasmus riescono a rilasciare un gioco devo ammettere che non mi sento per nulla prolifico come quei ragazzi. Vedo la luce alla fine del tunnel per quanto riguarda la codifica, ma ho ancora un sacco di lavoro per la creazione e la progettazione di livelli e della grafica. Quando recentemente progettai il livello del bosco, probabilmente impiegai l'equivalente di circa 3 giorni lavorativi da 8 ore l'uno. Per i nuovi livelli andrò più veloce se riuscirò ad ottimizzare il flusso di lavoro, ma vorrei ideare almeno altri 10 livelli prima di pensare di rilasciare il gioco ufficialmente.

Chris: Puoi fare una supposizione su una eventuale data di rilascio ?

The Mole: ho intenzione di rilasciare una demo giocabile entro mese. Avrò tre livelli giocabili e sarà principalmente destinata ad individuare eventuali bug rimanenti che la community su AtariAge potrà riscontrare. Spero anche di trovare un aiuto nel tirare avanti mentre lavorerò sul finale del gioco. Con un po' di fortuna potrò concludere in tempo il gioco in modo da riuscire a mettere uno dei moduli SSS sotto il mio albero di Natale 😊.

Chris: hai già pensato o immaginato a qualcosa che puoi fare per il tuo prossimo progetto e se sì puoi cosa pensi di sviluppare?

The Mole: tu non vuoi ancora lasciarmi in pace 😊, ci sono molte cose che vorrei veramente fare. Mi piacerebbe fare un ottimo gioco di corse pseudo-3D nello stile di Pole Position, ma con un motore di rendering molto affascinante. Uno dei miei preferiti di tutti i tempi nel genere fu Pit Stop II sul C64. Questo potrebbe essere un gioco unicamente per la F18A/v99x8, però, non sono sicuro che il VDP originale sarebbe in grado di supportarlo.

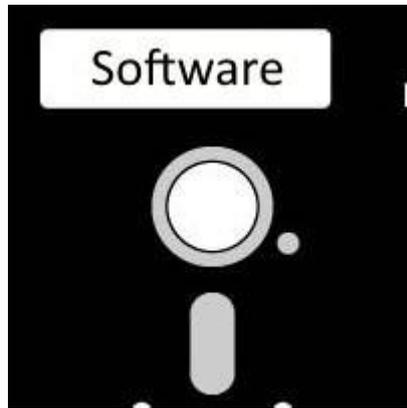
Non vedo l'ora di fare una porting per il gioco Ghostbusters di Activision. Questo è un altro dei miei giochi preferiti che è stato convertito per quasi tutti gli altri home computer e console e non c'è davvero ragione per cui il TI non potesse avere anche la sua. Questo è attualmente il gioco favorito per il mio prossimo progetto.

Infine, spero che io possa continuare nel far evolvere il motore di Alex Kidd con altre funzioni come lo scrolling orizzontale/verticale, su piani in pendenza, ecc. e di utilizzarlo per molti altri giochi. Alla fine spero che anche altre persone potranno usufruire dei miei tools e riuscire a progettare i propri giochi. Non sarà per i deboli di cuore e sarà un po' più complicato rispetto a strumenti come 'Magellan', ma spero di riuscire a renderli sufficientemente semplici in modo da motivare i non programmatori ad utilizzarlo.

[f]

E' accaduto a Settembre:

- **"The TI-99 High Score Contest"** organizzato su Atari Age da *Owen Brand* ha visto come protagonista il gioco "Miner2049er". Il vincitore si porta a casa un modulo Football del TI-99/4A incluso il relativo manuale. Il vincitore del mese scorso, "globeron", ha segnato in Munchman oltre 150.000 punti!!
- Ricordati di tenere d'occhio il nuovo gioco **"Bouncy"** di *Rasmus* (in fase di sviluppo). Sembra che sia brillante alla stregua dei suoi precedenti e spettacolari giochi.
- *Vorticon* sta attualmente sviluppando un nuovo gioco in *TurboForth* intitolato **"JetPac"**. Puoi leggere tutta la fase di sviluppo di questo gioco in AtariAge TI-99/4A Development forum.
- *Jedimatt42* sta lavorando su un adattatore USB, basato su Arduino, per usare una tastiera standard PC sul nostro TI-99 senza disabilitare la tastiera originale. Ci sono già progressi significativi. Visitate la sua pagina in AtariAge TI-99/4A Development forum.



[Un Menu stile Geneve 9640 per il nostro TI-99/4A](#)

Sembra che *Tim T. (InsaneMultitasker)* sia riuscito a fare un gorsso regalo a tutti noi! Infatti è riuscito a trovare il codice sorgente originale della versione a 80 colonne del BOOT del Geneve. Questo programma non fu mai distribuito pubblicamente e Tim sta lavorando per metterlo a punto.

(Puoi approfondire questo argomento sulla pagina AtariAge all'indirizzo di seguito:
<http://atariage.com/forums/topic/241745-80-column-menu/>)

Questo menu sicuramente soddisferà sia gli utenti della F18A che quelli del 9938. Il nuovo BOOT ti catapulta in un vero schermo ad 80 colonne nel quale vengono mostrate alcune selezioni (selezionabili con lettere dalla A alla X) ed sarà anche possibile concatenare i sistemi di menu salvandoli come eseguibili configurabili nel menu come selezioni per offrirti una quantità illimitata di opzioni. Questo è noto come concatenamento e può essere configurato con l'opzione 'F4' nel menu.

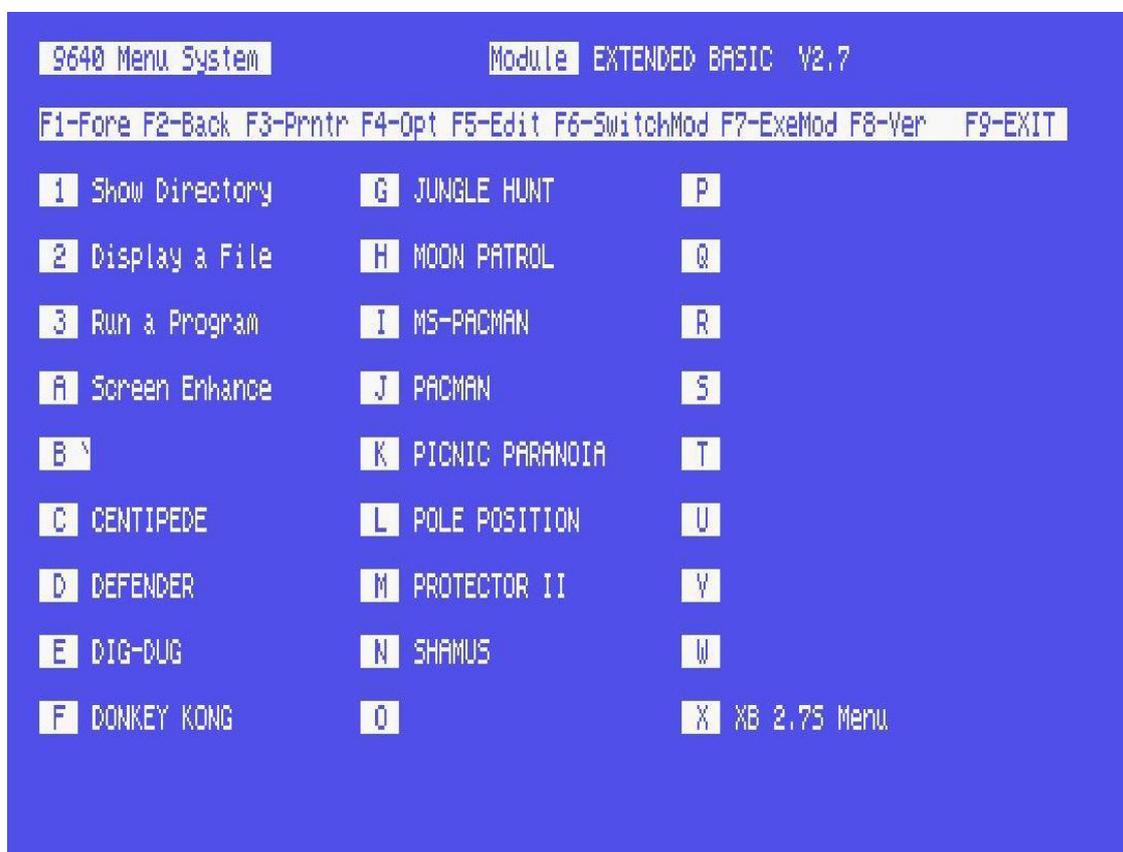
Questo nuovo sistema di menu supporta più schede di clock. E' stato testato anche con il l'emulatore 'MESS' con montata la scheda HSGPL. Basta essere sicuri di aggiornare la versione di MESS alla 0.153, che ha corretto vari bug per poterlo funzionare correttamente.

In ordine, per creare correttamente un concatenamento, occorre fare un po' di pianificazione. Per configurare il mio menu a catena, ho settato la concatenazione a mio piacimento chiamandolo, per esempio, 'Disk Utilities'. Dopo l'ho salvata e poi ho premuto F4 per salvarlo come eseguibile. Questo ha permesso il salvataggio del file su un floppy disk con il nome 'USEROPT1'. Potrai poi rinominare questo file tramite un programma di Disk Manager in modo da poter cambiare il nome del menu come si desidera. Continua a ripetere questa operazione per tutti i menu della concatenazione.

Una volta che avete tutti i menu concatenati è possibile modificare il menu ancora una volta e chiamare tutti i menu concatenati abbinati ai tasti di scelta settati. Qui sotto si possono vedere un paio di screenshot. Ne sono stati configurati una manciata per fornire qualche esempio:



la mia schermata del menu principale per il nuovo BOOT



esempio di un menu Disk Utility.

Naturalmente, tutti a conoscenza dei giochi *Atari* che compaiono nel menu!

Ho anche configurato un menù *TeleComm* per tutti i miei programmi di telecomunicazioni. Sto davvero solo testando le opzioni concatenamento e, come ha detto *Tim*, avete opzioni illimitate per i menu. Sono sicuro che non avrei potuto in alcun modo configurarne così tanta roba senza questo strumento. La maggior parte di noi probabilmente avrebbe potuto cavarsela con un solo menu per tutti i programmi che in genere usate o almeno aggiungere un menu per i giochi Atari, altri giochi e utilità. Penso che dal momento che ci sono così tanti giochi di Atari per il TI questi dovrebbero avere un proprio menu! Ho dedicato molte ore notturne per giocare con quei giochi quando ero ragazzo!

A mio parere questo sistema di menu è di gran lunga il più utile mai creato per il TI-99/4A. Ho sempre desiderato un sistema di menu come quello sul Geneve 9640 e ora sta diventando una realtà per gli utenti del TI-99/4A con le 80 colonne.



Games

Chiamata a **TUTTI I GIOCATORI!**

Owen Brand (Opry99er) ha avviato un concorso **TI Gaming** su AtariAge dove sul TI-99/4A viene scelto un gioco ogni mese e i Tlers possono competere per vedere chi può raggiungere il punteggio maggiore. A fine mese la persona con il punteggio più alto riceve un qualche tipo di premio.

Se vuoi leggere l'intera pagina del messaggio ed, eventualmente, partecipare amichevolmente alla competizione, allora clicca qui sotto:
<http://atariage.com/forums/topic/241547-official-ti-994a-hi-score-competition/page-1>

Il gioco del mese scorso era: *MunchMan*

Il vincitore è stato "*globeron*" con un punteggio di 153.840 punti.

Tutto quello che ho da dire è "**ACCIDENTI!**", deve avere proprio delle vesciche sulle mani!

Il gioco di questo mese è: **Miner 2049er**

Non ho mai giocato molto a Miner 2049er da bambino, ma di recente ho provato a vedere fin dove potevo arrivare. Questo gioco è molto più difficile di MunchMan! Sono riuscito ad arrivare al Terzo livello e ad ho raggiunto poco più di 9000 punti, devo ancora capire come si supera questo il terzo livello 😊.

Per aiutarci, Sometimes99er ha modificato il file .bin originale per aiutarci. Nel nuovo file modificato ci sono vite infinite ! Ricordate che questo file serve solo per capire il gioco e Sometimes99er ne ha cambiato il Font del punteggio per assicurarci che non ci sarà nessun imbroglio. È possibile trovare questo file nel post originale cliccando sul link che trovate sopra indicato.

BUONA FORTUNA!



Ho avuto più di una richiesta per avviare nella newsletter un tutorial del linguaggio assembly. Ho contattato un paio di persone che hanno esperienza nella programmazione con il linguaggio assembly del TMS9900. *Lee Stewart* è stato più che disponibile a dare una mano per questo primo tutorial. Naturalmente gli esercizi non dovranno estendersi troppo per evitare alla newsletter di diventare troppo lunga. Quindi abbiamo puntato su poche e brevi esercitazioni, solo per mostrare alcune operazioni di base. Spero che questo sia il primo di molti tutorial per aiutare gli utenti TI.

[Assembly Tutorial #1 – Screen Color di Lee Stewart](#)

In Questo Primo Tutorial, si vuole dimostrare come cambiare il colore dello schermo usando la programmazione assembler TMS9900, ma partiamo descrivendo alcune aree che questo tutorial andrà a coprire.

La cpu TMS9900 usa 3 registri di lavoro RAM-based, 3 interni e 16 generici. Il linguaggio assembler (ALC – Assembler Language Code) permette di lavorare direttamente su questi registri. Può anche prenotare e trasferire le informazioni in qualsiasi posizione di lavoro desiderata.

Informazioni maggiori riguardo il TMS9900 puoi trovarle all'indirizzo [Wikipedia](#).

[Per quanto riguarda questa parte, non essendo io \(ancora\) un programmatore assembler mi limiterò a riportare la parte in inglese originale. In caso ci fosse qualche volontario che si voglia occupare di tradurre questa sezione mi contatti pure.](#)

Workspace registers save time and instruction space and some instructions will only work with registers. Most CPUs have all of their registers on the CPU chip. The TMS9900 processor has only three 16-bit, on-chip registers: **PC** (*Program Counter*—contains the 16-bit address of the next instruction to execute), **WP** (*Workspace Pointer*—contains the CPU address of the first register of the current, 16-register, 32-byte workspace) and **ST** (*STatus*—contains the status result of the last instruction affecting status—not all do).

Registers are referenced by number: **0 – 15**. We can represent numbers in ALC by naming assembly-time constants with the **EQU** directive that make it easier for us to make sense of our program. That is why we usually refer to register #0 as R0, etc. Of course, we can use any name we want for a given constant:

```
R0 EQU 0
R1 EQU 1
.
.
.
R15 EQU 15
```

Actually, we do not need to write the above EQUates into our program because we can direct the Assembler to do it for us.

There are 5 kinds of addressing used by the TMS9900:

1. *Workspace Register,*
2. *Workspace Register Indirect,*
3. *Symbolic Memory,*
4. *Indexed Memory and*
5. *Workspace Register Indirect Auto-increment*

Four of the above use registers and only two use memory addresses directly. Our little SCREENCOLOR program uses only two, workspace-register and symbolic-memory addressing. There are many examples of workspace-register addressing in our program. Symbolic-memory addressing uses '@' in front of the address value or symbol. The only example of symbolic-memory addressing we use in our program is @VWTR.

One more thing before we get to the program: Hexadecimal (base 16) numbers are represented in TMS9900 ALC prefixed with '>', e.g., >01F0, which is 496 decimal (base 10).

Below is the code used to change the screen color (comments included):

```
*=== SCREENCOLOR=====
* This program changes the screen to TEXT mode without clearing
* the screen. The text from the Editor/Assembler "Load and Run"
* screen remains displayed but is skewed because the screen is 8
* characters wider.
*=====
REF VWTR      *reference E/A utility
DEF START    *declare Start label of our program so E/A loader
              * can find program entry point
              * Program entry point
```

START LI R4,>F485 *load R4 with pattern for color changes
* rotating bit pattern begins with 1111010010000101
*: LI = Load Immediate value into register. The immediate value must be known
*: at assembly time, so it can also be a label and involve a calculation.

LI R1,>0700 *load R1 with # of VDP register for screen color in MSB

LI R0,>01F0 *load R0 with TEXT mode setting for VDP register# 1
* VWTR needs R0 with MSB=VR#=(>01) and

LSB=setting=(>F0)

BLWP @VWTR *change to TEXT mode

*: BLWP = Branch and Load Workspace Pointer. Uses the workspace pointed to by

*: the source address/register; saves the current PC, WP and ST in the new R13,

*: R14 and R15; and branches to the address pointed to by the address 2 bytes

*: after the source address/register.

*: VWTR points to the WP and address of the E/A utility for loading values

*: into VDP registers.

* Infinite color change loop

NEXT LI R5,>6000 *load delay counter

MOV R4,R0 *copy R4 (contains changing FG-BG color byte in LSB) to R0

*: MOV = MOVE (copy) contents of source address/register to destination address/register

MOVB R1,R0 *copy MSB (>07) from R1 to MSB of R0 to change screen colors

*: MOVB = MOVE Byte from MSB of source address/register to MSB of destination address/register

BLWP @VWTR *change FG and BG screen color (FG only affects TEXT mode)

* VWTR needs R0 with MSB=VR#7 and LSB=colors

* colors are left nybble=FG; right nybble=BG

* Delay loop

DELAY NOP *a do-nothing operation

DEC R5 *DECRement delay count by 1

JNE DELAY *if R5<>0, continue delay loop

*: JNE = Jump to label if status of last instruction is Not Equal to 0

SRC R4,1 *shift R4 circularly right 1 bit to get different colors in LSB

*: SRC = Shift Right Circular the value in the register by the number of bits

*: indicated by the immediate value after the comma.

JMP NEXT *loop for next go-round

*: JMP = Jump unconditionally to label

END START *END of program; start automatically at START when program loaded

*== NOTES =====

== Labels start in first column. If '' is in column one, entire line is a comment.

*==

*== First text after spaces on non-comment line is an instruction word.

*==

*== After next space are one or more operands separated by commas with no

*== spaces (except within TEXT quotes).

*==

*== After next space, remainder of line is comment.

*==

*==ASM994A: ';' starts a comment anywhere on a line.

*==

*== Labels may be longer than 6 characters

*==

*== Program may be in lower case.

Regarding the values >0700 and >01F0 used early in the program, these have to do with the ALC function VWTR provided by the E/A cartridge. The name of the function stands for “VDP RAM Write Register”. That function requires a value in R0 before it is called. The MSB (Most Significant Byte [left byte]) of the value is the VDP register number (0 – 7) and the LSB (Least Significant Byte [right byte]) is the value to be written to that register—the registers are byte registers. >0700 has 7 in the MSB in preparation for getting it into the MSB of R0 later. We are preparing to write to VR#07, the screen-color register, a value for foreground (FG) color of text-mode characters and screen color (also, background [BG] color of text-mode characters). The FG and BG colors are 4-bits (1 nybble or ½ byte) each, e.g., >F4 is white (>F) on dark blue (4).

The value >01F0 is stashed in R0 prior to calling VWTR to write the value >F0 (LSB) to VR#01 (MSB) to change to text mode. Bit >10 of VR#01 is the text-mode-setting bit. See §21.1 of the E/A Manual for more information.

I hope this helps a lot of the ‘Assembly Beginners’. If anyone wants a certain assembly tutorial covered, please e-mail me directly. I cannot guarantee that it can be done in a newsletter as we are gearing this for beginners and covering somewhat very small assembly programs just to provide a basic understanding.

RESOURCES



Informazioni

Per contattarmi non esitate a visitare il mio sito e fare clic sulla scheda '[Contatti](#)'.

Argomenti per la Newsletter

Se volete partecipare alla stesura di questa newsletter e fornire argomenti per questa newsletter vi prego di contattarmi tramite il mio sito web.

Siti

Qui di seguito trovate le risorse in una manciata di siti che supportano i computer TI-99/4A e/o Geneve 9640. Non è certamente un elenco completo. Questa sezione sarà inclusa e aggiornata in tutte le prossime newsletter.

Siti Web / siti FTP

<http://www.ti99iuc.it>

<http://www.atariage.com>

<http://shift838.wix.com/shift838>

<http://www.99er.net>

<http://www.harmlesslion.com>

<http://www.mainbyte.com>

<http://www.ninerpedia.org/>

<http://www.ti99-geek.nl/>

<http://www.turboforth.net/>

<ftp://ftp.whtech.com>

<http://www.ti99hof.org/index.html>

<http://www.ti99ers.org/unsung/>

<http://ti99ers.org/modules/Inspire/remember.htm>

contiene tutti gli storici TI-99ers che sono deceduti.

Lista Gruppi Yahoo

<https://groups.yahoo.com/neo/groups/TI99-4A/info>

<https://groups.yahoo.com/neo/groups/TI994A/info>

<https://groups.yahoo.com/neo/groups/Geneve9640/info>

<https://groups.yahoo.com/neo/groups/turboforth/info>

BBS active

HeatWave BBS

Accesso: Dial-Up e Telnet

Sistema: Geneve 9640

Software: S&T BBS Software

Località: Arizona

Contenuto: biblioteche di file TI e Geneve, messaggi di base, porte giochi ed e-mail.

Telnet: www.heatwavebbs.com port 9640 dialup: 602-955-4491 @ 8-N-1

The Reef Hidden

Accesso: Dial-Up

Sistema: TI-99/4A modificato

Software: S&T BBS Software

Località: New York

Contenuto: biblioteche di file TI e Geneve, messaggi di base, porte giochi ed e-mail.

The Keep

Accesso: HTTP e Telnet

Sistema: Pentium 4 con sistema operativo Windows 2000

Software: Worldgroup BBS Software (fino a 256 connessioni utente)

Località: Tigard, Oregon

Contenuto: biblioteche di file TI e Geneve, messaggi di base, porte giochi, multi-utente e giochi multiplayer ed e-mail.

Telnet: www.thekeep.net porta 23 Web browser per <http://www.thekeep.net>

The Keep dispone di librerie di file TI, messaggi di base, e-mail, giochi porte, multi-utente e giochi multiplayer. The Keep ha anche una linea modem collegata con tutti coloro che desiderano contattare The Hidden Reef BBS da internet attraverso The Keep.

Semplicemente Telnet alla www.thekeep.net sulla porta 23, accedi a The Keep e quindi digita **/GO DIALOUT** nel menu principale, quindi D1 per la composizione verso The Hidden Reef. E' molto semplice.

Venditori

SHIFT838 – Fornisce componenti TI usati come li ha acquistati. Controllate spesso cosa ho disponibile. Un sacco di articoli possono essere riutilizzati da altri utenti TI.

Arcade Shopper - fornisce attrezzature TI vecchie e nuove, aggiornamenti e nuove piste PCB a www.arcadeshopper.com

Centri di riparazione

Richard Bell

Riparazioni disponibili su base limitata, si prega di contattare Richard a swim4home@verizon.net per conoscere i tempi di attesa prima di inviare qualsiasi componente da riparare

Tim

Riparazioni su hardware Myarc disponibili su base limitata. Contattare Tim a insane_m@hotmail.com per i tempi di attesa o per richiedere il servizio.