

# M.A.C. tutorials

## Guida alla programmazione di una avventura con il Mystery Adventure Creator.

### Introduzione.

Lo scopo di questo tutorial è mostrare la costruzione di una avventura testuale (*at* d'ora in poi) attraverso il Mystery Adventure Creator. Non è detto che sia il metodo seguito sia il migliore, ne quello più semplice o efficiente, è semplicemente quello con cui mi sono trovato meglio. E' indirizzato in particolare a coloro che sono al loro primo approccio da autori di *at* ed a digiuno di programmazione. Perciò chi ha un minimo lavorato con *mac* o con altri programmi di interactive fiction lo troverà assolutamente inutile e ridondante, ciononostante sono stato spinto a scriverlo dalle molte richieste che mi sono giunte in tal senso dopo la pubblicazione del principe dei ladri. Nell'esposizione del tutorial costruirò una piccola avventura sfruttando alcune delle possibilità offerteci dal m.a.c. non mostrerò tutte le possibili funzioni, ne quelle già presenti sui diversi esempi già disponibili. Semplicemente inserirò solo quegli elementi funzionali all'idea per questa mini avventura di esempio. Darò per scontata la lettura del manuale di Paolo Lucchesi.

### Step 1.

Creiamo nella posizione che più ci aggrada, una cartella con il nome di *macTutorial*, qui poniamo una copia del file *mactxt.exe* ed una copia del file *ital.adh* associati nel pacchetto zip ufficiale di *mac*. Rinomianiamo ora il file *mactxt.exe* in *macTutorial.exe*, e creiamo con un editor di testo qualsiasi un file chiamato *macTutorial.adv* (posto sempre nella cartella *macTutorial*).

*NOTA:* Il file \*.adv necessario per la creazione di una avventura con m.a.c. è un semplice file di testo, potrete utilizzare qualsiasi editor di testo. Dall'*edit.exe* del DOS al notepad di windows o al wordpad o anche il word di office (sconsigliato); Se decidete però di creare una avventura piuttosto ampia potrebbe convenire ricorrere ad un editor adatto alla programmazione come [editplus](#) o [ultraedit](#). Questi ultimi hanno la possibilità di visualizzare con colori diversi le parole chiave della programmazione facilitando l'analisi dello script. [Qui](#) potete trovare la personalizzazione per *mac* del wordfile.txt per *ultraedit* e le istruzioni per utilizzarlo, nel caso sceglieste questo editor.

*NOTA:* Per gli autori non vedenti si consiglia di utilizzare il file *Macstd.exe* al posto el *Mactxt.exe* poichè più adatto al riconoscimento dei programmi di videolettura.

Dal manuale del m.a.c. avrete appreso che lo script \*.adv è diviso in varie sezioni, ognuna di esse cura un aspetto della vostra avventura. Apriamo perciò il file macTutorial.adv che adesso è completamente bianco e scriviamo le varie sezioni:

```
[System]
[Shows]
[Locations]
[verbs]
[nouns]
[objects]
[messages]
[vars]
[Actions]
[Events]
```

In ognuna di esse andremo a porre i comandi che riteniamo più opportuni nella creazione della nostra avventura. Cominciamo perciò ad arricchire il nostro script. Limitiamoci come inizio a voler creare solo la descrizione di una stanza. Ecco cosa aggiungeremo al nostro script:

```
| inizio script
| il carattere "|" segnala la presenza di un
commento, la riga che inizia con "|" verrà
ignorata dal m.a.c. e serve solo per
| spiegare i comandi che la seguono
||| importiamo il file ital.adh:
@ital.adh

[System]
||| ogni oggetto, nome, locazione, verbo in
m.a.c. viene numerato per essere
identificato
||| impostiamo come locazione di partenza la
locazione n° 1:
StartLocation=1
||| impostiamo a 5 il numero massimo di
oggetti che può essere trasportato
contemporaneamente dal giocatore:
MaxItems=5

[Shows]
[Locations]
||| ora creiamo la nostra prima stanza, sarà
una semplice stanza vuota, senza alcuna
uscita:
#1,stanza vuota,,0
Ti trovi in una stanza vuota dalle pareti
bianche.

[verbs]
[nouns]
[objects]
[messages]
[vars]
[Actions]
[Events]
| fine script
```

Salviamo il file macTutorial.adv .

Importiamo il file `ital.adh` poichè contiene alcune impostazioni di default per m.a.c. tra cui la gestione dei messaggi di errore, il movimento con le direzioni cardinali, ed i verbi fondamentali delle at (come `esamina`, `prendi`, `lascia`, `salva` etc...). Al momento ci basti sapere questo, poi, più avanti, vedremo di modificarlo per renderlo più adatto ai nostri scopi. La sintassi per la definizione della locazione segue la seguente logica (attenzione alle virgole che sono molto importanti):

#numero identificativo, descrizione breve, nomefileimmagine, stanza illuminata, stanza a nord di questa, a sud, ad est , ad ovest, sopra, sotto  
descrizione lunga della locazione

se uno degli argomenti manca viene considerato con valore 0, come nel caso del tutorial, dove mancano il nome del file grafico associato alla locazione e l'indicazioni per le locazioni adiacenti alla locazione numero 1. Un ultima cosa a proposito dell'esempio, lo 0 posto subito dopo la virgola del filename, corrisponde al tipo di stanza: 0 se la stanza è illuminata ed 1 se la stanza è buia. Vedremo meglio questa impostazione quando tratteremo il problema dell'illuminazione degli ambienti.

A questo punto salviamo il file `macTutorial.adv` e lanciamo il file `macTutorial.exe`.

NOTA: il file `macTutorial.exe` cercherà automaticamente il file `macTutorial.adv` ; è importante che i due file siano nella stessa cartella ed abbiano lo stesso nome. Se non avete rinominato i file in tal modo allora dovrete utilizzare il prompt di dos (o il menu avvio di windows) digitando "...percorso/mactxt.exe vostro tutorial.adv".

Clicca [qui](#) per vedere l'immagine.

Avremo solo la descrizione della stanza, potremo salvare e caricare partite, consultare l'inventario (per ora vuoto), tentare di muoverci (tutte azioni previste dalla libreria), ma a qualunque altro comando riceveremo la deludente risposta "Non capisco quello che intendi". Questo risultato naturalmente non può soddisfarci. Innanzi tutto vorremmo aggiungere un messaggio iniziale con il titolo dell'avventura ed in secondo luogo vorremo consentire qualche azione al giocatore. Aggiungiamo perciò qualche riga al nostro script:

```
| inizio script
@ital.adh

[System]
StartLocation=1
MaxItems=5

[Shows]
||| inseriamo un messaggio di tipo show con
il titolo dell'avventura e l'autore.
||| si ricorda che il carattere "\" posto
all'interno del messaggio manda a capo il
testo nel gioco.
#1,
Benvenuti\\TUTORIAL per M.A.C\\scritto
da riktik\\| lascio qualche riga vuota

[Locations]
#1,stanza vuota,,0
Ti trovi in una stanza vuota dalle pareti
bianche.

[verbs]
```

#### [nouns]

||| aggiungiamo qualche nome "sensibile"

#100,STANZA

||| i nomi scritti sulla stessa riga (stesso numero) divisi da virgole sono considerati tutti sinonimi dal m.a.c.

#101,PARETI,MURI,PARETE,MURO

#### [objects]

#### [messages]

||| aggiungiamo un paio di messaggi nel caso il giocatore voglia guardarsi intorno

#1,La stanza è completamente vuota.

#2,Le pareti sono perfettamente bianche.

#### [vars]

#### [Actions]

||| aggiungiamo le azioni necessarie al fine di associare i messaggi alle azioni  
esamina/guarda-nome sensibile

if at 1

{+

if verb ESAMINA noun

STANZA

then message 1

if verb ESAMINA noun

PARETE

then message 2

}

#### [Events]

||| diciamo a mac di stampare sullo schermo

il titolo dell'avventura

if begin

then show 1 halt

| fine script

Salviamo il file macTutorial.adv.

Quando scriviamo la descrizione di una locazione, dovremo aggiungere alla sezione "noun" tutti quelle parole che il giocatore considera rilevanti e con cui potrebbe voler interagire. Nella nostra prima locazione diamo davvero poche informazioni al giocatore. Gli diciamo che si trova in una STANZA vuota dalle PARETI bianche. In questa situazione è probabile che il giocatore cerchi di approfondire la propria conoscenza dell'ambiente cercando di esaminare quei pochi indizi che gli abbiamo fornito. Abbiamo perciò aggiunto un paio di messaggi che verranno stampati dal gioco nel momento in cui il giocatore cercherà di esaminare la stanza o le pareti. Un giocatore con un poco di fantasia potrebbe immaginare che trovandosi in una stanza vi sia anche una porta e forse una finestra, sicuramente un pavimento ed un soffitto, per ora però supporremo che il giocatore sia un poco tardo e vada alla ricerca solo di ciò che vede :-). Avrete notato che ho numerato i due nomi a partire dal numero 100 e non dal numero 1, spigherò tale motivo quando inseriremo il primo oggetto nel gioco. Diamo invece un'occhiata alla sezione Actions, ecco come m.a.c. interpreta lo script: se il giocatore si trova nella

locazione n° 1 allora esegui la prima istruzione utile all'interno del blocco e fermati. Nel blocco, se il verbo scritto dal giocatore è "esamina" ed il complemento è "stanza" allora stampa il messaggio n°1 se invece il complemento è "parete" allora stampa il messaggio n°2. Qualsiasi altro comando non previsto darà la solita frase "Non capisco quello che intendi". Abbiamo creato il blocco "locazione 1" per il semplice motivo che potrebbero esserci altre stanze più avanti nel gioco nelle quali potremmo volere che alla stessa frase "esamina parete" sia data una risposta diversa, non tutte le stanze infatti possono essere uguali. Nella sezione Events invece abbiamo detto a mac che all'inizio del gioco vogliamo che stampi il messaggio di tipo show n° 1.

NOTA: M.A.C. riconosce tre tipi di blocchi: { } {+ } {/ } la differenza tra i tre è ben spiegata nel manuale. Qui ho utilizzato il secondo tipo perchè più adatto alla condizione "at 1".

NOTA: abbiamo utilizzato il messaggio di tipo show per il titolo, ma in realtà non avendo posto alcuna immagine avremmo potuto utilizzare al posto di questo un semplice messaggio di testo.

Lanciamo ancora una volta il file macTutorial.exe.

Clicca [qui](#) e [qui](#) per vedere le immagini.

Adesso il giocatore vedrà il titolo dell'avventura e potrà esaminare la stanza e le pareti, ottenendo risposte assolutamente inutili. Anch'esse però contribuiscono al gioco, infatti il giocatore comincerà a farsi un'idea di desolazione e comincerà a pensare di essere prigioniero nella stanza, visto che non riesce ad uscirne. Se vogliamo che la nostra avventura sia decente, non possiamo fermarci qui, dobbiamo prevedere una quantità di verbi e di azioni che il giocatore utilizzerà per interagire con lo scarso ambiente che lo circonda. La maggior parte di esse potrebbe dare risposte infruttuose (come abbiamo visto), ma altre provocheranno qualche evento che favorirà lo sviluppo del gioco, ma di questo parleremo nel prossimo step.

## STEP 2.

Per migliorare l'interazione del nostro gioco dovremmo un poco aggiornare il vocabolario di m.a.c. , cominciamo dai verbi. Al momento gli unici verbi che abbiamo definito sono quelli presenti nel file *ital.adh* (apritelo e dategli un'occhiata). Facciamo perciò una lista di tutti i verbi utili che potrebbero essere utilizzati dal giocatore, non preoccupatevi se ne dimenticate qualcuno potrete sempre aggiornarla in seguito. In più scriviamo un messaggio di default per ogni verbo. Cos'è un messaggio di default? E' il messaggio che sarà mostrato durante il gioco quando il giocatore associerà quel verbo ad un "noun" che non abbiamo previsto. Ad esempio se tra i verbi abbiamo previsto ROMPI ed il giocatore cercasse di rompere qualunque cosa gli venisse in mente, invece di prevedere ogni possibile azione "Rompi" "oggetto" potremmo stampare un messaggio di default del tipo "la violenza è inutile in questo caso". Per ragioni di praticità (lo script diventerà sempre più lungo) sfrutteremo da subito il comando @ importa file. Ciò complicherà un poco l'apprendimento, ma ci metterà subito nella giusta ottica per creare una avventura degna di tal nome. Creeremo perciò tre file di testo \*.adh: uno contenente la sezione verbs (con l'elenco di tutti i verbi che intendiamo usare), un secondo file contenete le frasi di default, l'ultimo contenete le azioni necessarie per associare le frasi di default ai verbi. Infine li importeremo nel giusto posto dello script principale.

Creiamo il file di testo "*verb.adh*", sempre nella cartella macTutorial, e scriviamo al suo interno il seguente elenco di verbi:

### [Verbs]

#31,ACCENDI,BRUCIA,APPICCIA  
#32,ANNUSA  
#33,APRI  
#34,AIUTA  
#35,AIUTO  
#36,ASCOLTA,SENTI,AUSCULTA  
#37,AVVOLGI  
#38,BALLA,DANZA  
#39,BEVI  
#40,BUSSA  
#41,BUTTA,LANCIA,GETTA  
#42,CAGA  
#43,CANCELLA,ANNULLA  
#44,CANTA  
#45,CHIUDI  
#46,CHIEDI,DOMANDA  
#47,COLPISCI,AGGREDISCI,SCAGLIATI,ATTACCA,CACCIA  
#48,COMPRA  
#49,CONTA,NUMERA  
#50,DAI,CONSEGNA  
#51,DORMI,RIPOSA  
#52,ENTRA  
#53,ESCI  
#54,FERMA,ARRESTA,STOPPA  
#55,GIOCA  
#56,GIRA,RUOTA  
#57,GUARDATI,GUARDAMI  
#58,GUIDA,CONDUCI  
#59,INCOLLA,ATTACCA,UNISCI,COLLEGA  
#60,INFILA  
#61,INSULTA,OFFENDI,STUPIDO,IDIOTA  
#62,ISTRUZIONI,INFORMAZIONI  
#63,LIBERA  
#64,LECCA

#65,LEGA  
#66,LEGGI,SFOGLIA  
#67,MANGIA,ASSAGGIA,TRANGUGIA  
#68,MOSTRA  
#69,NASCONDITI,NASCONDIMI  
#70,PARLA,RISPONDI,DISCUTI  
#71,PERQUISISCI  
#72,PULISCI  
#73,PUNTEGGIO,PUNTI  
#74,RIEMPI,COLMA  
#75,ROMPI,FORZA,DISTRUGGI,DANNEGGIA,FRANTUMA,SPEZZA,  
BUCA  
#76,RUBA,SOTTRAI,SGRAFFIGNA,SOLA  
#77,SALI  
#78,SALUTA,ADDIO  
#79,SALTA,SCAVALCA,SUPERA,OLTREPASSA,BALZA  
#80,SCENDI  
#81,SCIOGLI,SLEGA,DIRIMI  
#82,SCRIVI,COMPILA,REDAI,DISEGNA,SCARABOCCHIA  
#83,SCUREGGIA  
#84,SEPARA,DIVIDI  
#85,SFILA  
#86,SPINGI,SPOSTA,MUOVI  
#87,SPEGNI  
#88,SPORCA,INSUDICIA  
#89,SPUTA,SCATARRA  
#90,SUCCHIA,ASPIRA  
#91,SVEGLIA  
#92,SVUOTA,VERSA,COLA  
#93,TAGLIA,SEGA,ABBATTI,STRAPPA  
#94,TIRA  
#95,TOCCA,TASTA,PALPA  
#96,TROMBA,SCOPA,FOTTI  
#97,TUFFATI,BUTTATI,LANCIATI  
#98,UCCIDI,ACCOPPA,AMMAZZA  
#99,UCCIDITI,SUICIDATI,MUORI  
#100,USA,UTILIZZA  
#101,VAI,TORNA,RAGGIUNGI  
#102,BACIA,LIMONA

|||fine verbi

Notate che abbiamo cominciato l'elenco dal numero 31, le prime trenta posizioni le lasceremo infatti libere per il file *ital.adh* (vedremo poi come modificarlo).

NOTA: ogni sezione (ad eccezione di *system*, *actions*, *events*) ha una numerazione propria è importante che nella medesima sezione non vi siano più righe con il medesimo numero. Ciò può accadere per sbaglio nel momento in cui si importano altri file con pezzi di codice, come noi ci accingiamo a fare. Per tale motivo lascerò sempre un commento nel codice principale che ricordi i numeri già occupati in altri file.

Salviamo il file *verb.adh* e creiamo un altro file chiamato "*msgdef.adh*" nel quale andremo a scrivere i messaggi di default dei vari verbi. Numereremo i messaggi a partire dal 500. Ecco le frasi di default che ho scritto per l'elenco dei miei verbi:

[Messages]

||| messaggi di default dei verbi

- #500,Non si può accendere.
- #501,Non senti nulla.
- #502,Non si può aprire.
- #503,Non ha bisogno di aiuto.
- #504,Silenzio assoluto.
- #505,Non puoi avvolgerlo.
- #506,Improvvisi un paio di passi di danza, col risultato di renderti ridicolo.
- #507,Non puoi berlo.
- #508,Toc Toc.
- #509,Non puoi liberartene così.
- #510,Sarebbe meglio trovare un bagno.
- #511,Non si cancella.
- #512,Sei stonato come una campana.
- #513,Non si può chiudere.
- #514,Chiedere non serve a nulla.
- #515,La violenza non è la risposta giusta.
- #516,Non è in vendita.
- #517,Inutile contare.
- #518,Sei troppo generoso.
- #519,Chi dorme non piglia pesci.
- #520,Non puoi entrare lì.
- #521,Uscire di lì non è possibile.
- #522,Non si può fermare.
- #523,Trastulli per fanciulli.
- #524,Non si gira.
- #525,Bello come il sole.
- #526,Non hai la patente.
- #527,Impossibile unirli.
- #528,Non si infila.
- #529,Questo è un gioco per bene.

||| inizio istruzioni

- #530,ISTRUZIONI\1>Avventure testuali\2>Comandi\3>Credits\4>Torna al gioco.\ |
- #531,AVVENTURE TESTUALI\La descrizione della vostra avventura\\(premi un tasto)\
- #532,COMANDI\La descrizione dei comandi principali\\(premi un tasto)\
- #533,CREDITS\L'autore della avventura ed un ringraziamento a Paolo Lucchesi\\(premi un tasto)\\

||| fine istruzioni

- #534,Non si può liberare.
- #535,Mettere la lingua ovunque non è bello.
- #536,Non si può legare.
- #537,Niente di interessante.
- #538,Non è il momento di mangiare.
- #539,Meglio non mostrarlo.
- #540,Non vedi posti in cui nascondersi.
- #541,Chiacchiere inutili.
- #542,Niente di particolare.
- #543,Non è il momento per le pulizie di pasqua.

||| punteggio:

- #544,Hai totalizzato %varPunti% Punti in %varTurni% mosse.
- #545,Non si può riempire.
- #546,La violenza non è tutto.
- #547,Questo è un gioco onesto.
- #548,Non si può salire lì.
- #549,Fai un gesto con la mano.
- #550,Non sei così atletico.
- #551,Non si può scendere di lì.
- #552,Non si può sciogliere.
- #553,Pessima grafia.



```

#554,Dovresti limitarti.
#555,Impossibile da dividere.
#556,Non si può sfilare.
#557,Non si sposta.
#558,Non si spegne.
#559,Non essere dispettoso.
#560,Splath....
#561,Bleah.
#562,Meglio lasciar stare.
#563,Non si può svuotare.
#564,Niente da fare.
#565,Non si muove.
#566,Niente di particolare al tatto.
#567,Questo è un gioco per bene.
#568,Non è il momento di fare acrobazie.
#569,Non vorrai finire in galera?
#570,Non ti abbattere.
#571,Sii più specifico.
#572,Andar di la non è possibile.
#573,Il bacio è un apostrofo rasa tra le parole "ti" ed "amo".
#574,Arrangiate.
||| fine messaggi di default

```

Salviamo il file msgdef.adh

Notate in questo script un messaggio in particolare, il numero 544. Esso presenta dei caratteri % al suo interno. In particolare questi caratteri contengono due parole: varPunti e varTurni queste sono due variabili numeriche che andremo più avanti a inserire nello script principale nella sezione [vars]. La prima misurerà i punti conquistati dal giocatore durante la partita, la seconda il numero di comandi da lui inseriti per raggiungere tale punteggio. Il punteggio torna utile al giocatore facendolo cosciente dei suoi progressi nell'avventura e quindi incoraggiandolo ad andare avanti.

Creiamo un altro file chiamato "*verbdef.adh*" nel quale andremo a scrivere le seguenti azioni che associano ad ogni verbo il corrispettivo messaggio di default:

```

[Actions]
if verb ACCENDI noun 0
then message 500
if verb ANNUSA noun 0
then message 501
if verb APRI noun 0
then message 502
if verb AIUTA noun 0
then message 503
if verb AIUTO noun 0
then message 574
if verb ASCOLTA noun 0
then message 504
if verb AVVOLGI noun 0
then message 505
if verb BALLA noun 0
then message 506
if verb BEVI noun 0
then message 507
if verb BUSSA noun 0
then message 508
if verb BUTTA noun 0
then message 509

```

```

if verb CAGA noun 0
then message 510
if verb CANCELLA noun 0
then message 511
if verb CANTA noun 0
then message 512
if verb CHIEDI anynoun 0
then message 513
if verb CHIUDI noun 0
then message 514
if verb COLPISCI noun 0
then message 515
if verb COMPRA noun 0
then message 516
if verb CONTA noun 0
then message 517
if verb DAI anynoun 0
then message 518
if verb DORMI noun 0
then message 519
if verb ENTRA noun 0
then message 520
if verb ESCI noun 0
then message 521
if verb FERMA noun 0
then message 522
if verb GIOCA noun 0
then message 523
if verb GIRA noun 0
then message 524
if verb GUARDA noun 0
then message 525
if verb GUIDA noun 0
then message 526
if verb INCOLLA noun 0
then message 527
if verb INFILA noun 0
then message 528
if INSULTA verb noun 0
then message 529
||| ISTRUZIONI
if verb ISTRUZIONI noun 0
{

    set varIstruzioni 1
    while eq varIstruzioni 1
    {

        cls message 530 mchoiche varR 4
        if eq varR 1
        then cls message 531 pause
        if eq varR 2
        then cls message 532 pause
        if eq varR 3
        then cls message 533 pause
        if eq varR 4
        then cls set varIstruzioni 0
        reset varR

    }
}

```

```
}  
|||FINE ISTRUZIONI  
if verb LIBERA noun 0  
then message 534  
if verb LECCA noun 0  
then message 535  
if verb LEGA noun 0  
then message 536  
if verb LEGGI noun 0  
then message 537  
if verb MANGIA noun 0  
then message 538  
if verb MOSTRA anynoun 0  
then message 539  
if verb NASCONDITI noun 0  
then message 540  
if verb PARLA anynoun 0  
then message 541  
if verb PERQUISISCI noun 0  
then message 542  
if verb PULISCI noun 0  
then message 543  
if verb PUNTEGGIO noun 0  
then message 544  
if verb RIEMPI noun 0  
then message 545  
if verb ROMPI noun 0  
then message 546  
if verb RUBA noun 0  
then message 547  
if verb SALI noun 0  
then message 548  
if verb SALUTA noun 0  
then message 549  
if verb SALTA noun 0  
then message 550  
if verb SCENDI noun 0  
then message 551  
if verb SCIOGLI noun 0  
then message 552  
if verb SCRIVI noun 0  
then message 553  
if verb SCUREGGIA noun 0  
then message 554  
if verb SEPARA noun 0  
then message 555  
if verb SFILA noun 0  
then message 556  
if verb SPINGI noun 0  
then message 557  
if verb SPEGNI noun 0  
then message 558  
if verb SPORCA noun 0  
then message 559  
if verb SPUTA noun 0  
then message 560  
if verb SUCCHIA noun 0  
then message 561  
if verb SVEGLIA noun 0  
then message 562  
if verb SVUOTA noun 0  
then message 563
```

```

if verb TAGLIA noun 0
then message 564
if verb TIRA noun 0
then message 565
if verb TOCCA noun 0
then message 566
if verb TROMBA noun 0
then message 567
if verb TUFFATI noun 0
then message 568
if verb UCCIDI noun 0
then message 569
if verb UCCIDITI noun 0
then message 570
if verb USA noun 0
then message 571
if verb VAI noun 0
then message 572
if verb BACIA noun 0
then message 573
|||finedefault

```

Salviamo il file verbdef.adh

Il "noun 0" sta per nome sconosciuto o nessun nome. Avrete notato in quest'ultimo script la presenza di un'azione particolare ossia quella associata al comando istruzioni. Essa presenta un ciclo while, ed un dialogo a scelta multipla con l'utente. Prendiamolo come è per il momento, più avanti spiegheremo il significato in particolare di questa scrittura. (lo so, lo so, sto rimandando tante cose, ma abbiate pazienza).

Ricapitolando, adesso abbiamo, nella nostra cartella macTutorial, 6 files: macTutorial.exe , macTutorial.adv , ital.adh , verb.adh , msgdef.adh , verbdef.adh ; riprendiamo il file macTutorial.adv (nostro script principale) e vediamo come andare ad introdurre il lavoro che abbiamo fatto sui verbi:

```

| inizio script
@ital.adh

[System]
StartLocation=1
MaxItems=5

[Shows]
#1,
Benvenuti\\TUTORIAL per M.A.C\\scritto da riktik\\

[Locations]
#1,stanza vuota,,0
Ti trovi in una stanza vuota dalle pareti bianche.

|||al posto della sezione [verbs] importiamo il file verb.adh che la contiene.
@verb.adh

[nouns]
#100,STANZA
#101,PARETI,MURI,PARETE,MURO

```

#### [objects]

#### [messages]

#1,La stanza è completamente vuota.

#2,Le pareti sono perfettamente bianche.

||| aggiungiamo i messaggi di default associati ai verbi

||| lasciamo anche un commento a ricordarci che essi vanno dal n°500 al n°574

||| #500-inizio messaggi di default

@msgdef.adh

||| #574-fine messaggi di default

#### [vars]

||| definiamo le variabili che abbiamo utilizzato negli script importati (e che non abbiamo ancora spiegato)

varTurni

varPunti

varR

#### [Actions]

if at 1

{+

if verb ESAMINA noun STANZA

then message 1

if verb ESAMINA noun PARETE

then message 2

}

||| alla fine della sezione actions importiamo il file verbdef.adh

@verbdef.adh

#### [Events]

||| diciamo a mac di stampare sullo schermo il titolo dell'avventura

if begin

then show 1 halt

||| ad ogni turno incrementiamo la variabile che li conta:

add varTurni 1

| fine script

Salviamo il tutto e lanciamo il file macTutorial.exe.

Clicca [qui](#) e [qui](#) per vedere l'immagini.

Abbiamo inserito i verbi, i messaggi di default e la loro associazione. La posizione in cui vengono inseriti tali file è importante. In particolare ricordate che il file verbdef.adh deve rimanere sempre come ultima istruzione della sezione actions. Questo perchè le risposte di default vanno visualizzate solo nel momento in cui non abbiamo associata alcun altra risposta/azione migliore. Ad esempio provate a dare ora il comando "ascolta" avrete la risposta di default "Silenzio assoluto", ogni stanza in cui non sia prevista una diversa risposta ci darà quella di default. Ma se andiamo ad aggiungere un nuovo messaggio ed una nuova azione nella stanza uno avremo che il gioco al comando ascolta (nella stanza uno) ci darà la nuova risposta (e non stamperà quella di default, perchè successiva):

| inizio script  
@ital.adh

#### [System]

StartLocation=1  
MaxItems=5

#### [Shows]

#1,  
Benvenuti\\TUTORIAL per M.A.C\\scritto da riktik\\

#### [Locations]

#1,stanza vuota,,0  
Ti trovi in una stanza vuota dalle pareti bianche.

||| al posto della sezione [verbs] importiamo il file verb.adh che la  
contiene.

@verb.adh

#### [nouns]

#100,STANZA  
#101,PARETI,MURI,PARETE,MURO  
||| aggiungiamo il noun musica  
#102,MUSICA,MELODIA

#### [objects]

#### [messages]

#1,La stanza è completamente vuota.  
#2,Le pareti sono perfettamente bianche.  
||| aggiungiamo un messaggio relativo alla musica.  
#3,Senti una musica dolce ed armoniosa, ma non riesci a capire da  
dove proviene.  
||| #500-inizio messaggi di default  
@msgdef.adh  
||| #574-fine messaggi di default

#### [vars]

varTurni  
varPunti  
varIstruzioni  
varR

#### [Actions]

if at 1  
{+  
  
    if verb ESAMINA noun STANZA  
    then message 1  
    if verb ESAMINA noun PARETE  
    then message 2  
    if verb ASCOLTA noun 0  
    or verb ASCOLTA noun MUSICA  
    then message 3  
  
}

||| alla fine della sezione actions importiamo il file verbdef.adh  
@verbdef.adh

#### [Events]

```

if begin
then show 1 halt

add varTurni 1

| fine script

```

Adesso al comando "ascolta" o "ascolta la musica" verrà visualizzato il messaggio numero 3 e non quello di default.

Ma ora, come promesso, occupiamoci delle variabili: m.a.c. riconosce unicamente variabili di tipo numeriche, esse possono poi contenere solo numeri positivi interi. Attraverso speciali comandi (li trovate nel manuale) è possibile variarne o controllarne il valore. Possono essere utilizzate per contare i punti ed i turni, ma il loro scopo principale è quello di fungere da flag (bandiere), segnali per dare il via, o consentire, una certa azione prima preclusa. Il loro utilizzo sarà più chiaro nel momento in cui inseriremo qualche enigma nell'avventura. Le variabili che vengono utilizzate vanno dichiarate nella sezione vars. Per praticità ed onde evitare qualsiasi confusione con nomi ed oggetti porrò davanti ad ogni variabile il suffisso var (come nello script). Al momento ho dichiarato 4 variabili. La variabile varTurni ha il compito di contare il numero di mosse del giocatore, essa incrementa di una unità ad ogni turno (sezione events); La variabile varPunti deve invece contare il numero di punti accumulati dal giocatore, al momento questa variabile non viene modificata nello script e rimane pari a 0 (valore iniziale di ogni variabile se non diversamente specificato); entrambi i valori possono essere visualizzati dal giocatore dando il comando "punteggio" (grazie al messaggio n°544 come abbiamo visto precedentemente). La variabile varR è stata utilizzata (per la scelta multipla) assieme alla variabile varIstruzioni (che funge da flag) al fine di gestire un menù esplicativo del gioco diviso in più sezioni, dove ognuna di esse è raggiungibile premendo il numero associato ad essa. Andiamo a vedere e a spiegare il ciclo while e il dialogo a scelta multipla usato all'interno del file verbdef.adh.

```

| se il giocatore dà il comando "istruzioni"
if verb ISTRUZIONI noun 0
| allora
{

    | impostiamo il valore di varIstruzioni
    (la nostra flag) pari ad 1
    set varIstruzioni 1
    | mentre il valore di varIstruzioni è pari
    ad 1
    while eq varIstruzioni 1
    | esegui le seguenti istruzioni e ripetile
    fin quando varIstruzioni sarà pari ad 1
    {

        | pulisci lo schermo e stampa il messaggio n° 530 (tale messaggio contiene i
        titoli delle sezioni del menu istruzioni ognuno associato ad un numero da uno
        a quattro) chiedi al giocatore un numero da uno a quattro ed assegna tale
        valore alla variabile varR (non accettare altri comandi all'infuori dei numeri da
        uno a quattro):
        cls message 530 mchoiche varR 4
        | se la risposta del giocatore è la uno (associata nel messaggio 530 alla
        descrizione dell'avventura) allora pulisci lo schermo stampa il messaggio 531
        (descrizione dell'avventura) e aspetta che l'utente prema un pulsante per

```

```

        proseguire.
        if eq varR 1
        then cls message 531 pause
        | se la risposta del giocatore è 2 allora pulisci lo schermo e stampa il
        messaggio 532 (descrizione dei comandi) e aspetta che il giocatore prema un
        pulsante qualsiasi per proseguire
        if eq varR 2
        then cls message 532 pause
        | se la risposta è 3 mostra i crediti
        if eq varR 3
        then cls message 533 pause
        | Se la risposta è quattro (torna al gioco nel messaggio 530) allora imposta la
        variabile varIstruzioni uguale a 0
        if eq varR 4
        then cls set varIstruzioni 0
        | poni a 0 la variabile varR
        reset varR

        | fine ciclo, se la condizione varIstruzioni è pari ad 1 allora
        ricomincia altrimenti esci dal ciclo e fermati
    }

}

```

Da qui si evince l'importanza delle variabili come flags, senza il controllo operato sulla variabile varIstruzioni non sarebbe possibile uscire dal ciclo while ed esso si ripeterebbe all'infinito. La pausa impostata dopo la stampa dei messaggi a schermo serve a fermare il ciclo prima che esso riparti pulendo lo schermo e chiedendo ancora un numero da 1 a 4. Questo sistema può essere utilizzato anche per creare dei dialoghi a scelta multipla con altri personaggi non giocatori come avviene nelle avventure grafiche, anche se non è un metodo utilizzato di frequente nelle at.

### Step 3.

Aggiungiamo qualche optional al nostro gioco. Inseriremo un comando StatusLine al fine di avere sempre in primo piano il numero di mosse ed il punteggio, in più aggiungeremo un messaggio che appaia ogni tanto sullo schermo mentre il giocatore è nella stanza n° 1.

```

| inizio script
@ital.adh

[System]
StartLocation=1
MaxItems=5
||| come potete vedere segue la stessa logica del messaggio associato all'azione
"punteggio"
StatusLine=Hai realizzato %varPunti% punti in %varTurni% mosse.

[Shows]
#1,
Benvenuti\\TUTORIAL per M.A.C\\scritto da riktik\\

[Locations]
#1,stanza vuota,,0
Ti trovi in una stanza vuota dalle pareti bianche.

```



```
|||al posto della sezione [verbs] importiamo il file verb.adh che la contiene.  
@verb.adh
```

#### [nouns]

```
#100,STANZA  
#101,PARETI,MURI,PARETE,MURO  
||| aggiungiamo il noun musica  
#102,MUSICA,MELODIA
```

#### [objects]

#### [messages]

```
#1,La stanza è completamente vuota.  
#2,Le pareti sono perfettamente bianche.  
||| aggiungiamo un messaggio relativo alla musica.  
#3,Senti una musica dolce ed armoniosa, ma non riesci a capire da dove proviene.  
||| #500-inizio messaggi di default  
@msgdef.adh  
||| #574-fine messaggi di default
```

#### [vars]

```
varTurni  
varPunti  
varIstruzioni  
varR
```

#### [Actions]

```
if at 1  
{+  
  
    if verb ESAMINA noun STANZA  
    then message 1  
    if verb ESAMINA noun PARETE  
    then message 2  
    if verb ASCOLTA noun 0  
    or verb ASCOLTA noun MUSICA  
    then message 3  
  
}
```

```
||| alla fine della sezione actions importiamo il file verbdef.adh  
@verbdef.adh
```

#### [Events]

```
if begin  
then show 1 halt
```

```
add varTurni 1
```

```
||| frase random  
if at 1 random 1 8  
then message 3  
| fine scrip
```

Clicca [qui](#) per vedere l'immagine.

Aggiungiamo ancora qualche frase di contorno, personalizziamo al gioco la risposta ai tentativi (inutili) di movimento nella stanza n°1:

```
| inizio script
@ital.adh
```

#### [System]

```
StartLocation=1
MaxItems=5
StatusLine=Hai realizzato %varPunti% punti in %varTurni% mosse.
```

#### [Shows]

```
#1,
Benvenuti\\TUTORIAL per M.A.C\\scritto da riktik\\
```

#### [Locations]

```
#1,stanza vuota,,0
Ti trovi in una stanza vuota dalle pareti bianche.
```

```
|||al posto della sezione [verbs] importiamo il file verb.adh che la contiene.
@verb.adh
```

#### [nouns]

```
#100,STANZA
#101,PARETI,MURI,PARETE,MURO
#102,MUSICA,MELODIA
||| aggiungiamo due noun
#103,PAVIMENTO
#104,SOFFITTO
```

#### [objects]

#### [messages]

```
#1,La stanza è completamente vuota.
#2,Le pareti sono perfettamente bianche.
#3,Senti una musica dolce ed armoniosa, ma non riesci a capire da dove proviene.
||| aggiungiamo un paio di messaggi.
#4,Il pavimento ed il soffitto, come le pareti, sono perfettamente bianchi. Se non fosse per la
gravità sarebbe impossibile stabilire quale sia il sopra ed il sotto.
#5,Non sembra esservi modo di uscire di qui.
#6,Le pareti sono solide!
#7,Solido!
#8,Troppo duro da rompere a mani nude!
#9,Fai forza con tutto il tuo peso, ma non si muove di un millimetro.
||| #500-inizio messaggi di default
@msgdef.adh
||| #574-fine messaggi di default
```

#### [vars]

```
varTurni
varPunti
varIstruzioni
varR
```

#### [Actions]

```
if at 1
{+

    ||| aggiungiamo qualche azione "inutile"
    if verb ESAMINA noun PAVIMENTO
    or verb ESAMINA noun SOFFITTO
    then message 4
    if verb TOCCA noun PAVIMENTO
    or verb TOCCA noun SOFFITTO
    then message 7
```

```

    if verb TOCCA noun PARETI
    then message 6
    if verb ROMPI noun PARETI
    or verb ROMPI noun PAVIMENTO
    or verb COLPISCI noun PARETI
    or verb COLPISCI noun PAVIMENTO
    then message 8
    if verb SPOSTA noun PARETI
    then message 9
    if verb ESAMINA noun STANZA
    then message 1
    if verb ESAMINA noun PARETE
    then message 2
    if verb ASCOLTA noun 0
    or verb ASCOLTA noun MUSICA
    then message 3
    ||| movimenti:
    if move
    or verb VAI noun 0
    then message 5

}

||| alla fine della sezione actions importiamo il file verbdef.adh
@verbdef.adh

[Events]
if begin
then show 1 halt

add varTurni 1

if at 1 random 1 8
then message 3
| fine script

```

Oltre ad alcune risposte ad azioni ‘inutili’ abbiamo aggiunto anche un messaggio casuale, l’istruzione è l’ultima della sezione events, ad ogni turno ed indipendentemente dal comando del giocatore con una probabilità di 1 su 8 il gioco stamperà il messaggio numero tre informando il giocatore di una misteriosa melodia...

Ora che abbiamo un poco più di confidenza andiamo ad aggiungere alla nostra stanza un oggetto. Un oggetto si differenzia da un semplice "noun" per il fatto che m.a.c. ne gestisce automaticamente alcune proprietà come il suo trasporto, l’inventario, la descrizione, se sia o meno una fonte luminosa, se può essere indossato... Il numero identificativo di un oggetto è strettamente correlato con il numero identificativo di un nome, ossia l'oggetto n° 1 sarà associato automaticamente al nome n°1 (con qualche eccezione che vedremo più avanti, non odiateci). Ecco perchè nella sezione nouns abbiamo iniziato la numerazione da 100, ho infatti riservato le prime 100 posizioni ad eventuali oggetti (ma non ho intenzione di farne tanti, tranquilli). I nomi che ho definito fino ad ora, non avranno un oggetto corrispondente, infatti sarebbe del tutto inutile definire tanti oggetti, quando la loro funzione è unicamente quella di scenografia. Aggiungerò allora un oggetto chiamato "sfera" con il numero identificativo 1 ed il corrispondente noun 1 = "sfera". La nostra sfera sarà un oggetto particolare, non potrà infatti essere raccolta ed è un oggetto luminoso.

NOTA: Utilizzo un oggetto per la sfera (e non la includo nella descrizione) poichè sarà possibile eliminarla dalla stanza. Se avessi incluso la sfera nella descrizione della

stanza, alla sua eliminazione, avrei dovuto cambiare anche la descrizione della locazione, per fare ciò avrei dovuto creare una nuova locazione con la medesima descrizione ad eccezione della sfera e trasferirvi il giocatore. Un lavoro assai dispendioso quando si può invece sfruttare le caratteristiche degli oggetti.

| inizio script  
@ital.adh

#### [System]

StartLocation=1  
MaxItems=5  
StatusLine=Hai realizzato %varPunti% punti in  
%varTurni% mosse.

#### [Shows]

#1,  
Benvenuti\\TUTORIAL per M.A.C\\scritto da riktik\\

#### [Locations]

#1,stanza vuota,,0  
Ti trovi in una stanza vuota dalle pareti bianche.

||al posto della sezione [verbs] importiamo il file  
verb.adh che la contiene.

@verb.adh

#### [nouns]

||| nouns associati ad oggetti  
#1,SFERA,BOCCIA,PALLA  
||| altri nouns  
#100,STANZA  
#101,PARETI,MURI,PARETE,MURO  
#102,MUSICA,MELODIA  
#103,PAVIMENTO  
#104,SOFFITTO

#### [objects]

||| ecco il nostro primo oggetto  
#1, la sfera,1,4  
La sfera è sospesa a circa un metro e mezzo da terra,  
emana uno strano bagliore fluorescente che illumina tutta  
la stanza.

#### [messages]

#1,La stanza è completamente vuota.  
#2,Le pareti sono perfettamente bianche.  
#3,Senti una musica dolce ed armoniosa, ma non riesci a  
capire da dove proviene.  
||| aggiungiamo un paio di messaggi.  
#4,Il pavimento ed il soffitto, come le pareti, sono  
perfettamente bianchi. Se non fosse per la gravità sarebbe  
impossibile stabilire quale sia il sopra ed il sotto.  
#5,Non sembra esservi modo di uscire di qui.  
#6,Le pareti sono solide!  
#7,Solido!  
#8,Troppo duro da rompere a mani nude!  
#9,Fai forza con tutto il tuo peso, ma non si muove di un  
millimetro.  
#10,E' troppo grande per metterla in tasca, e poi una volta  
presa non sapresti dove andare...  
||| #500-inizio messaggi di default

```
@msgdef.adh
||| #574-fine messaggi di default
```

```
[vars]
varTurni
varPunti
varIstruzioni
varR
```

```
[Actions]
```

```
if at 1
{+
```

```
||| aggiungiamo una risposta diversa a quella standard se il giocatore tenta di prendere
la sfera
```

```
if verb PRENDI noun SFERA present 1
then message 10
if verb ESAMINA noun PAVIMENTO
or verb ESAMINA noun SOFFITTO
then message 4
if verb TOCCA noun PAVIMENTO
or verb TOCCA noun SOFFITTO
then message 7
if verb TOCCA noun PARETI
then message 6
if verb ROMPI noun PARETI
or verb ROMPI noun PAVIMENTO
or verb COLPISCI noun PARETI
or verb COLPISCI noun PAVIMENTO
then message 8
if verb SPOSTA noun PARETI
then message 9
if verb ESAMINA noun STANZA
then message 1
if verb ESAMINA noun PARETE
then message 2
if verb ASCOLTA noun 0
or verb ASCOLTA noun MUSICA
then message 3
||| movimenti:
if move
or verb VAI noun 0
then message 5
```

```
}
```

```
||| alla fine della sezione actions importiamo il file
verbdef.adh
```

```
@verbdef.adh
```

```
[Events]
```

```
if begin
then show 1 halt
```

```
add varTurni 1
```

```
if at 1 random 1 8
then message 3
```

```
| fine script
```

Clicca [qui](#) per vedere l'immagine.

Notate l'inserimento della condizione "present 1" questo ci permetterà di evitare risposte inadeguate nel caso il giocatore cerchi di prendere la sfera anche quando essa non è presente. Parliamo ora, però, della problematica dell'illuminazione nei giochi creati con m.a.c. (ed in generale di tutte le at). [Probabilmente non è l'approccio migliore agli oggetti, sarebbe stato meglio vedere prima le altre proprietà più semplici]

NOTA: per la sintassi completa degli oggetti, consultate il manuale.

Spesso nelle at ci si trova all'interno di ambienti chiusi, qui sorge la problematica dell'illuminazione degli stessi, sia per una questione di verosimiglianza del mondo costruito, sia al fine di sfruttare tale possibilità per la creazione o complicazione di alcuni enigmi. M.a.c. ci aiuta a gestire tali situazioni attraverso l'impostazione di alcuni parametri. Nella definizione della locazione ricorderete che abbiamo impostato a 0 il parametro "tipo di stanza", esso indicava che la locazione è illuminata, diversamente il valore 1 indica al gioco che la locazione è buia, a meno che non vi sia una fonte di luce (oggetto luminoso). Cosa significa che la locazione è buia? Significa che m.a.c. sostituirà alla descrizione della stanza quella di locazione buia presente nel file ital.adh e solo quando in essa vi sarà un oggetto luminoso (o noi cambieremo l'impostazione della locazione) mostrerà la descrizione della locazione posta nello script principale. Rileggendo queste righe non ci ho capito niente neanche io, perciò, onde spiegarmi, andiamo a vedere il caso pratico: cambiamo il parametro nella locazione 1 affinché venga considerata locazione buia (se non vi sono fonti di luce). Ed aggiungiamo la possibilità di togliere la sfera dalla stanza (eliminando in tal modo la fonte di luce). All'inizio perciò il gioco considererà la locazione come buia, ma essendovi la sfera (oggetto luminoso), la locazione apparirà come in precedenza (illuminata) poi dando l'opportuno comando per togliere la sfera, essa verrà eliminata e con essa l'illuminazione della locazione.

```
| inizio script
@ital.adh
```

```
[System]
StartLocation=1
MaxItems=5
StatusLine=Hai realizzato %varPunti% in punti %varTurni% mosse.
```

```
[Shows]
#1,
Benvenuti\\TUTORIAL per M.A.C\\scritto da riktik\\
```

```
[Locations]
||| cambiamo il tipo di locazione (da illuminata a buia)
#1,stanza vuota,,1
Ti trovi in una stanza vuota dalle pareti bianche.
```

```
|||al posto della sezione [verbs] importiamo il file verb.adh che la contiene.
@verb.adh
```

```
[nouns]
||| nouns associati ad oggetti
#1,SFERA,BOCCIA,PALLA
||| altri nouns
#100,STANZA
#101,PARETI,MURI,PARETE,MURO
#102,MUSICA,MELODIA
#103,PAVIMENTO
#104,SOFFITTO
```

### [objects]

#1, la sfera,1,4

La sfera è sospesa a circa un metro e mezzo da terra, emana uno strano bagliore fluorescente che illumina tutta la stanza.

### [messages]

#1,La stanza è completamente vuota.

#2,Le pareti sono perfettamente bianche.

#3,Senti una musica dolce ed armoniosa, ma non riesci a capire da dove proviene.

#4,Il pavimento ed il soffitto, come le pareti, sono perfettamente bianchi. Se non fosse per la gravità sarebbe impossibile stabilire quale sia il sopra ed il sotto.

#5,Non sembra esservi modo di uscire di qui.

#6,Le pareti sono solide!

#7,Solido!

#8,Troppo duro da rompere a mani nude!

#9,Fai forza con tutto il tuo peso, ma non si muove di un millimetro.

#10,E' troppo grande per metterla in tasca, e poi una volta presa non sapresti dove andare...

#11,Spingi la sfera, ed essa, dopo una piccola, resistenza comincia a muoversi; acquistando velocità ti sfugge dalle mani e finisce contro la parete.\Incredibilmente essa non viene fermata dall'ostacolo, ma vi si tuffa dentro come una goccia nell'acqua e scompare lasciando sulla parete cerchi concentrici che subito però scompaiono.

||| #500-inizio messaggi di default

@msgdef.adh

||| #574-fine messaggi di default

### [vars]

varTurni

varPunti

varIstruzioni

varR

### [Actions]

if at 1

{+

||| aggiungiamo la possibilità di eliminare la sfera

if verb SPINGI noun SFERA present 1

then message 11 destroy 1

if verb PRENDI noun SFERA present 1

then message 10

if verb ESAMINA noun PAVIMENTO

or verb ESAMINA noun SOFFITTO

then message 4

if verb TOCCA noun PAVIMENTO

or verb TOCCA noun SOFFITTO

then message 7

if verb TOCCA noun PARETI

then message 6

if verb ROMPI noun PARETI

or verb ROMPI noun PAVIMENTO

or verb COLPISCI noun PARETI

or verb COLPISCI noun PAVIMENTO

then message 8

if verb SPOSTA noun PARETI

then message 9

if verb ESAMINA noun STANZA

then message 1

if verb ESAMINA noun PARETE

then message 2

if verb ASCOLTA noun 0

```

        or verb ASCOLTA noun MUSICA
        then message 3
        ||| movimenti:
        if move
        or verb VAI noun 0
        then message 5

    }

    ||| alla fine della sezione actions importiamo il file verbdef.adh
    @verbdef.adh

    [Events]
    if begin
    then show 1 halt

    add varTurni 1

    if at 1 random 1 8
    then message 3
    | fine script

```

Clicca [qui](#) e [qui](#) per vedere le immagini.

Ora, spingendo la sfera, essa viene eliminata e subito la stanza si abbuia. Notate che siete sempre nella locazione n°1, è solo la descrizione della stanza ad essere sostituita, potrete perciò compiere ancora le azione previste in questo luogo. Per aggiungere veridicità al gioco, però, dobbiamo modificare lo script. E' infatti incoerente lasciare la possibilità di esaminare l'ambiente quando questo è buio. Perciò:

```

| inizio script
@ital.adh

[System]
StartLocation=1
MaxItems=5
StatusLine=Hai realizzato %varPunti% in punti %varTurni% mosse.

[Shows]
#1,
Benvenuti\\TUTORIAL per M.A.C\\scritto da riktik\\

[Locations]
||| cambiamo il tipo di locazione (da illuminata a buia)
#1,stanza vuota,,1
Ti trovi in una stanza vuota dalle pareti bianche.

|||al posto della sezione [verbs] importiamo il file verb.adh che la contiene.
@verb.adh

[nouns]
||| nouns associati ad oggetti
#1,SFERA,BOCCIA,PALLA
||| altri nouns
#100,STANZA
#101,PARETI,MURI,PARETE,MURO
#102,MUSICA,MELODIA
#103,PAVIMENTO
#104,SOFFITTO

```



### [objects]

#1, la sfera,1,4

La sfera è sospesa a circa un metro e mezzo da terra, emana uno strano bagliore fluorescente che illumina tutta la stanza.

### [messages]

#1,La stanza è completamente vuota.

#2,Le pareti sono perfettamente bianche.

#3,Senti una musica dolce ed armoniosa, ma non riesci a capire da dove proviene.

#4,Il pavimento ed il soffitto, come le pareti, sono perfettamente bianchi. Se non fosse per la gravità sarebbe impossibile stabilire quale sia il sopra ed il sotto.

#5,Non sembra esservi modo di uscire di qui.

#6,Le pareti sono solide!

#7,Solido!

#8,Troppo duro da rompere a mani nude!

#9,Fai forza con tutto il tuo peso, ma non si muove di un millimetro.

#10,E' troppo grande per metterla in tasca, e poi una volta presa non sapresti dove andare...

#11,Spingi la sfera, ed essa, dopo una piccola, resistenza comincia a muoversi; acquistando velocità ti sfugge dalle mani e finisce contro la parete.\Incredibilmente essa non viene

fermata dall'ostacolo, ma vi si tuffa dentro come una goccia nell'acqua e scompare lasciando sulla parete cerchi concentrici che subito però scompaiono.

#12,E' troppo buio per scorgere alcunché.

#13,Procedi a tentoni, fino a scontrarti contro la parete.

#14,Silenzio assoluto.

||| #500-inizio messaggi di default

@msgdef.adh

||| #574-fine messaggi di default

### [vars]

varTurni

varPunti

varIstruzioni

varR

### [Actions]

if at 1

{+

||| se la stanza è buia, allora è logico che non si possano guardare gli oggetti

||| è importante che il comando preceda gli altri "esamina":

if verb ESAMINA dark

then message 12

if move dark

then message 13

if verb ASCOLTA noun 0 dark

or verb ASCOLTA noun MUSICA dark

then message 14

if verb SPINGI noun SFERA present 1

then message 11 destroy 1

if verb PRENDI noun SFERA present 1

then message 10

if verb ESAMINA noun PAVIMENTO

or verb ESAMINA noun SOFFITTO

then message 4

if verb TOCCA noun PAVIMENTO

or verb TOCCA noun SOFFITTO

then message 7

if verb TOCCA noun PARETI

then message 6

if verb ROMPI noun PARETI

```

    or verb ROMPI noun PAVIMENTO
    or verb COLPISCI noun PARETI
    or verb COLPISCI noun PAVIMENTO
    then message 8
    if verb SPOSTA noun PARETI
    then message 9
    if verb ESAMINA noun STANZA
    then message 1
    if verb ESAMINA noun PARETE
    then message 2
    if verb ASCOLTA noun 0
    or verb ASCOLTA noun MUSICA
    then message 3
    ||| movimenti:
    if move
    or verb VAI noun 0
    then message 5

}

||| alla fine della sezione actions importiamo il file verbdef.adh
@verbdef.adh

[Events]
if begin
then show 1 halt

add varTurni 1

if at 1 random 1 8
then message 3
| fine script

```

A questo punto lo script comincerà ad essere un poco troppo lungo per essere seguito su questa pagina html, in più oramai si sarà presa confidenza col mezzo. Perciò da qui in poi metterò i collegamenti allo script macTutorial.adv nella sua evoluzione. Prima di ogni collegamento inserirò una descrizione di ciò che verrà aggiunto, mentre all'interno degli script troverete i commenti alle nuove linee inserite.

#### Step 4.

Inserita una introduzione all'avventura più corposa.  
 Clicca [qui](#) per lo script.

#### Step 5.

Inseriti alcuni miglioramenti all'ambiente e modifichiamo il file ital.adh con l'inserimento di alcune shortcut molto diffuse nella maggiorparte delle at.

Clicca [qui](#) per lo script adv e [qui](#) per lo script adh modificato.

#### Step 6.

Cambio di locazione. Inserito un personaggio non giocatore.  
 Clicca [qui](#) per lo script.

### **Step 7.**

Inserimento di un dialogo con un personaggio non giocatore.  
Clicca [qui](#) per lo script.

### **Step 8.**

Alcuni migliramenti qui e li. Clicca [qui](#) per lo script.

### **Step 9.**

Concludiamo l'avventura. Clicca [qui](#) per lo script.

### **Step 10.**

Eccoci dunque giunti alla conclusione di questo tutorial, questi sono i file completi utilizzati: [mactutorial.adv](#) [verb.adh](#) [verbdef.adh](#) [msgdef.adh](#) e [ital.adh](#). Mentre [qui](#) potete trovare il sorgente riunito in un unico file. E [qui](#) potete ottenere l'avventura d'esempio compilata e pronta per essere giocata. Adesso tocca a voi! Create la vostra avventura!!!

### *POST SCRIPTUM*

*Se avete suggerimenti o rilevate errori in questo tutorial, fatemelo sapere: [riktik@tiscalinet.it](mailto:riktik@tiscalinet.it)*