

DOOPSI_Docs

COLLABORATORS

	TITLE : DOOPSI_Docs		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		June 5, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	DOOPSI_Docs	1
1.1	DOOPSI-GS - DOOPSI Game System	1
1.2	DOOPSI-GS - Introduzione	3
1.3	DOOPSI-GS - Autori	4
1.4	DOOPSI-GS - Ringraziamenti e Saluti	5
1.5	DOOPSI-GS - DISCLAIMER	6
1.6	DOOPSI-GS - Copyrights	7
1.7	DOOPSI-GS - Installazione	7
1.8	DOOPSI-GS - Finestra Principale	9
1.9	DOOPSI-GS - Note	12
1.10	DOOPSI-GS - Path Editor	13
1.11	DOOPSI-GS - Path Tutorial 1	14
1.12	DOOPSI-GS - Path Tutorial 2	15
1.13	DOOPSI-GS - Descrizione Generale del Path Editor	17
1.14	DOOPSI-GS - Spot Editor	19
1.15	DOOPSI-GS - Spot Tutorial 1	20
1.16	DOOPSI-GS - Descrizione Generale dello Spot Editor	20
1.17	DOOPSI-GS - Attributi della Scena	21
1.18	DOOPSI-GS - Attributi degli Oggetti	22
1.19	DOOPSI-GS - Attributi di un Oggetto: Tutorial 1	22
1.20	DOOPSI-GS - Attributi di un Oggetto: Tutorial 2	23
1.21	DOOPSI-GS - Descrizione Generale degli Attributi di un Oggetto	23
1.22	DOOPSI-GS - DOOPSI Coder	25
1.23	DOOPSI-GS - Coder Tutorial 1	26
1.24	DOOPSI-GS - Coder Tutorial 2	27
1.25	DOOPSI-GS - Descrizione Generale del Coder	29
1.26	DOOPSI-GS - Anim Workshop	29
1.27	DOOPSI-GS - Anim Workshop Tutorial 1	30
1.28	DOOPSI-GS - Anim Workshop Tutorial 2	30
1.29	DOOPSI-GS - Descrizione Generale di Anim Workshop	31

1.30 DOOPSI-GS - Dialog Editor	33
1.31 DOOPSI-GS - Dialog Editor Tutorial 1	34
1.32 DOOPSI-GS - Descrizione Generale del Dialog Editor	35
1.33 DOOPSI-GS - Preferenze	37
1.34 DOOPSI-GS - Appendice A	39
1.35 DOOPSI-GS - La Console	46
1.36 DOOPSI-GS - Dentro DOOPSI	47
1.37 DOOPSI-GS - Il Player	48
1.38 DOOPSI GS - Player Royalty	50
1.39 DOOPSI-GS - Limiti	51
1.40 DOOPSI-GS - Shareware	52

Version 1.10

Introduzione	Cos'è DOOPSI?
Autori	I cervelli dietro il capolavoro.
Ringraziamenti e Saluti	Alcuni ringraziamenti.
DISCLAIMER	Licenza di distribuzione.
Copyrights	DOOPSI-GS (C) Copyrights
Shareware Info	DOOPSI-GS è Shareware!
Installazione	Come installare DOOPSI a mano.

DOOPSI Editor

Finestra Principale	Finestra Principale dell'Editor.
Path Editor	Come creare i percorsi.
Tutorial 1	
Tutorial 2	
Descrizione	
Spot Editor	Per manipolare gli Spot.
Tutorial 1	
Descrizione	
Attributi Schermo	Per cambiare gli attributi della Scena.
Attributi Oggetti	Per modificare gli attributi degli Oggetti.
Tutorial 1	
Tutorial 2	
Descrizione	
DOOPSI Coder	Per definire il comportamento degli Oggetti.
Tutorial 1	
Tutorial 2	
Descrizione	
Anim Workshop	Per creare animazioni.
Tutorial 1	
Tutorial 2	
Descrizione	
Editor Dialoghi	Facciamoci una chiacchierata!
Tutorial 1	
Descrizione	
Preferenze	I colori che preferite.

APPENDICI

Appendice A	Istruzioni DOOPSI
Appendice B	La Console del Player
Appendice C	Dentro DOOPSI
Limiti	Cosa potete e non potete fare.

DOOPSI PLAYER

Il Player	Il grande Player DOOPSI
Alcune Note	Dovreste leggere questo.
Player Royalty	Per distribuire il Player.

1.2 DOOPSI-GS - Introduzione

```
*****  
Introduzione.  
*****
```

Il nome del gioco.

Doopsi-GS significa: Dinamical Object Oriented Programming System Interface - Game System. Abbiamo dato questo nome al nostro programma per vari motivi: ci ricorda Boopsi, ovvero l'interfaccia Intuition Orientata agli Oggetti del nostro beneamato Amiga; è un programma Orientato agli Oggetti nel senso tecnico del termine, cioè per quanto riguarda la sua struttura interna (qualche notizia di questa struttura è riportata nell'Appendice C); è un'Interfaccia di Programmazione poiché semplifica il compito di programmazione rendendolo completamente automatico e veloce.

Questo Sistema di Programmazione è molto diverso dagli altri prodotti dedicati al medesimo scopo poiché è estremamente semplice da usare, anche per coloro che avessero poca abilità di programmazione: l'utente disegna tutta la grafica e fornisce tutti i suoni e le musiche di cui necessita e poi il nostro programma mette tutto assieme. L'idea che giace sotto il funzionamento di questo programma è immediata: voi definite prima di tutto le scene, poi ponete alcuni oggetti in esse e, infine, specificate come ciascun oggetto deve reagire quando il giocatore agisce su di esso (questa è l'unica parte di programmazione che dovete affrontare). Per essere sincero vi sono altre parti del lavoro che non ho incluso nella descrizione precedente, ma le imparerete man mano che verrete maggiormente coinvolti in Doopsi: per esempio, uno dei compiti che dovete affrontare è quello di costruire il percorso lungo il quale l'Omino cammina (l'Omino con la maiuscola è il protagonista dell'avventura), oppure un'altro compito è quello di scrivere i dialoghi fra l'Omino e qualche altro oggetto, e così via.

Queste sono alcune caratteristiche di Doopsi:

- sensibile al sistema operativo (funziona sotto V37+).
 - completamente localizzabile (usa la nostra "locator.library" per cambiare linguaggio, ma nelle future versioni lo adatteremo alla standard "locale.library" se riusciremo a farla funzionare sotto V37).
 - compatibile AGA.
 - l'Editor è completamente system friendly.
 - gestisce bitmap più grosse dell'area di schermo (con autoscroll).
-

- permette di suonare moduli e campionamenti durante il gioco.
- completamente personalizzabile (ad es.: potete anche ridisegnare la pulsantiera del Player).

Doopsi è costituito da due programmi indipendenti: l'Editor ed il Player. Come i loro nomi possono suggerire, l'Editor è quello che costruisce l'avventura, mentre il Player sfrutta i dati prodotti dall'Editor per farvi giocare. E' come se l'Editor generasse una specie di programma ed il Player fosse l'interprete di tale programma: se si verifica un errore durante l'esecuzione il Player visualizza un messaggio di errore e, se può, continua l'elaborazione del gioco, come un qualsiasi interprete di un qualunque linguaggio farebbe (i linguaggi reali però non possono continuare dopo un errore, ma noi siamo migliori e possiamo!).

Il manuale ha la seguente struttura: questa sezione introduttiva è seguita dalla descrizione della procedura di installazione. Poi iniziamo con la Finestra Principale dell'Editor Doopsi, contenente la descrizione di tutti i (numerosi) bottoni che appaiono (non vi spaventate, imparerete presto ad usarli). Dopo la Finestra Principale vi sono tre sezioni dedicate ai tre editor relativi alla costruzione della scena: il Path Editor, lo Spot Editor e lo Screen Attributes Editor. Seguono le sezioni dedicate agli oggetti e che descrivono rispettivamente: l'Object Attributes Editor, il Doopsi Coder e l'Anim Editor. Rimangono solo due editor, e questi costituiscono il soggetto delle sezioni seguenti: il Dialog Editor e l'Editor di Preferenze. La parte principale del manuale è finita ma abbiamo anche tre appendici: la prima descrive tutte le istruzioni del Linguaggio Doopsi (il Linguaggio Doopsi è usato nel Doopsi Coder per programmare gli oggetti); la seconda descrive il profilo della console inclusa nel pacchetto (la console è la parte bassa del gioco, quella con le azioni); e la terza contiene alcuni dettagli tecnici a proposito della struttura interna degli Oggetti Doopsi (questa non è necessaria ad una prima lettura, ma è inclusa per utenti più esperti (o più curiosi)). Anche il Player abbisogna di un minimo di spiegazione, e lo descriviamo nell'ultima sezione del manuale, situata dopo tutte le appendici. Questo è tutto, gente.

Una osservazione: probabilmente questo manuale è un po' caotico, nel senso che le informazioni utili sono sparpagliate ovunque e nascoste fra mucchi di rifiuti. Questo significa che dovete essere più attenti nel leggere, in modo da non saltare alcune nozioni interessanti (e vitali). Siamo spiacenti, ma questa è solo la prima versione e continueremo a sistemarla in modo da migliorarla. Dunque ogni commento è gradito, e terremo anche una lista di FAQ nelle nostre pagine Web.

1.3 DOOPSI-GS - Autori

```
*****
Gli autori (la parte più interessante del manuale):
*****
```


questa è la situazione fino al 27 - 6 - 1996 (data del primo upload):

Andrea si è appena laureato in Fisica all'Università di Milano, e sta studiando per entrare in un corso di Dottorato.

Fabio sta lavorando come capo progetto presso la DeAgostini Multimedia di Milano. Ama gli Amiga, i dolci, le ragazze, divertirsi, programmare a oggetti e DOOPSI ;)

I nostri indirizzi sono:

Fabio Rotondo	Andrea Galimberti
c.so Vercelli, 9	via Villioresi, 87
28100 Novara	20029 Turbigo (Mi)
Italia	Italia
E-Mail: fsoft@intercom.it	

Phone: (ITA) - (0)321 - 459676 Phone: (ITA) - (0)331 - 871009

Le pagine Web sono:

http://www.intercom.it/~fsoft/doorsi.html	-> DOOPSI Home Page!!!
http://www.intercom.it/~fsoft	-> Fabio Soft Home Page

DOOPSI Mailing list:

per iscriversi mandare una mail a:

fsoft@intercom.it

con:

SUBSCRIBE DOOPSI your_name@your_email.addr

nel corpo del messaggio.

1.4 DOOPSI-GS - Ringraziamenti e Saluti

Ringraziamenti:

Mik and Gio of ClassX: per alcune utili routine. Per il loro supporto,
 per le loro idee, per essere loro.

Giorgio Fornara: per la grandiosa grafica della Demo che non è inclusa
 in questo pacchetto, ma che apparirà il più presto
 possibile. Ha anche disegnato il logo di Doopsi che
 appare sullo schermo dell'Editor.

Andrea Rotondo: per la grafica del Tutorial.

Stefano Clemente: per il beta-testing.

Lele e Paolo di Intercom: per aver creato la DOOPSI - Mailing list.

Max Galimberti: per aver controllato questo manuale.

Saluti:

da Andrea a Laura: per avermi sopportato fino ad ora.

da Fabio alla sua mamma (e anche al suo papà).

1.5 DOOPSI-GS - DISCLAIMER

ACCORDO DI LICENZA PER DOOPSI-GS

I. DIRITTI E RESTRIZIONI DI QUESTA LICENZA

Fabio Rotondo e Andrea Galimberti (gli "autori") vi assicurano il diritto non-esclusivo, non-assegnabile di usare il software DOOPSI-GS incluso come eseguibile ("DOOPSI") su di un massimo di 1 computer system alla volta. Non potete effettuare reverse engineering, decompilare, o dissassemblare DOOPSI, non rispettare queste restrizioni e' espressamente proibito dalle leggi vigenti. Non potete prestare o affittare DOOPSI, o trasferire DOOPSI in altro modo e il materiale scritto che lo accompagna. Tutti i diritti non espressamente citati sono riservati dagli autori.

II. NESSUNA GARANZIA

DOOPSI e il materiale scritto che lo accompagna e' fornito "cosi' com'e'", senza alcuna garanzia di alcun tipo. Per quanto consentito dalla legge, gli autori scaricano tutte le responsabilita', sia implicite che esplicithe, comprese implicite garanzie di vendibilita', utilita' per un compito particolare e di non violazione. Gli interi rischi derivanti dall'uso o dal comportamento di DOOPSI e del materiale scritto che l'accompagna rimangono a voi.

III. NESSUN OBBLIGO PER DANNI CONSEGUENTI L'USO

Per quanto consentito dalla legge: in alcun caso gli autori o i loro collaboratori potranno essere accusati per qualsiasi danno (compresi, senza limitazione, danni per profitti di affari, interruzione degli affari, perdita di dati commerciali, o qualsiasi possibilita' di perdita monetaria) derivante dall'uso o dalla inadeguatezza di usare DOOPSI, persino se gli autori sono stati avvisati della possibilita' di tali danni. Dal momento che alcuni stati/giurisdizioni non permettono l'esclusione o la limitazione di garanzia per danni consequenziali o incidentali, questa limitazione potrebbe non riguardarvi.

IV. COPIA E DISTRIBUZIONE

Siete autorizzati a distribuire e copiare questo documento con tutto il resto dell'archivio, solo se il programma e' una versione "SHAREWARE" o "SPECIAL EDITION". La "SPECIAL EDITION" non puo' essere distribuita su Aminet o in qualsiasi BBS e non puo' comparire in alcuna rivista senza il permesso scritto degli autori. Voi potete, e siete invitati a farlo, copiare e distribuire la versione originale di DOOPSI SHAREWARE VERSION, ma NON DOVETE COPIARE in qualsiasi forma la DOOPSI REGISTERED VERSION (versione registrata), eccezion fatta per l'utilizzo personale.

V. VARIE

Se avete acquistato questo prodotto negli Stati Uniti, questo Accordo e' soggetto alle leggi dello Stato di Washington. Se avete acquistato questo prodotto fuori dagli Stati Uniti, verranno applicate le leggi locali. Se una qualsiasi parte esterna si arroca dei diritti derivanti dal presente Accordo, la parte prevalente potrebbe essere autorizzata a richiedere somme di denaro per il risarcimento per l'offesa, i costi e le spese.

1.6 DOOPSI-GS - Copyrights

Copyrights.

DOOPSI-GS e DOOPSI-Player sono (C)Copyright by Brigthing Brain Brothers

ReqTools è Copyright (c) Nico François e Magnus Holmgren.

1.7 DOOPSI-GS - Installazione

Installazione.

Vi è uno script per l'Installer che svolge questo compito, ma per coloro che desiderassero farlo a mano descriveremo cosa deve essere installato e dove. (A proposito: per usare lo script dovete possedere l'Installer e l'utility Lha.)

I tre archivi hanno già la struttura finale delle directory. Dovete prima estrarre l'archivio Programs.lha; quest'ultimo crea la directory Doopsi che contiene l'Editor ed il Player. Vi è anche una seconda directory in cui potete trovare due librerie: la "locator.library" e la famosa "reqtools.library"; dovete copiare queste librerie nella vostra directory Libs:. L'ultimo passo consiste nel fare la seguente assegnazione: Assign Doopsi: alla directory Doopsi.

La lingua di default è l'inglese, ma sono forniti anche i cataloghi per installare l'italiano: dovete solo estrarre l'archivio Catalogs.lha nella directory Doopsi (non NEL cassetto Doopsi, ma VICINO a tale cassetto). Questo crea una directory Catalogs nel cassetto Doopsi ed aggiunge anche il file "player.catalog" nella directory "Doopsi/Data".

Poi dovete estrarre l'archivio Tutorial.lha nella stessa directory dove avete messo il cassetto Doopsi (non NEL cassetto Doopsi, ma VICINO a tale cassetto): questo crea una directory Tutorial nel cassetto Doopsi, dove potete trovare un po' di grafica per iniziare immediatamente.

Per riassumere: gli archivi son stati costruiti in modo tale che, se li estraete tutti nella medesima directory, il loro contenuto finirà nei posti giusti.

La struttura risultante delle directory è:

```
DOOPSI (DIR)
|
+---Catalogs (DIR)
|
+---Docs      (DIR)
|
+---palettes  (DIR)
|
+---Data      (DIR)
|
'---Tutorial  (DIR)
    |
    +---Objects (DIR)
    |
    +---Berny   (DIR)
    |
    +---Scenes  (DIR)
    |
    '---Sounds  (DIR)
```

L'Editor Deve trovare nella propria directory il file "doopsi.dat". La directory Catalogs serve solo se volete usare un linguaggio diverso dall'inglese (per il momento sono forniti solo i cataloghi italiani): copiate in tale cassetto solo i cataloghi del vostro linguaggio preferito e lasciate fuori gli altri.

Il Player Deve trovare nella directory che scegliete mediante l'Editor (si veda la sezione di questo manuale dedicata alla finestra di Preferenze) i seguenti file: "player.dat" e "panel.iff". Il file "panel.iff" è la console che potete cambiare (vedere l'Appendice B). Il file "player.catalog" è necessario solo se volete usare un linguaggio diverso dall'inglese, altrimenti potete buttarlo via.

I sorgenti dei cataloghi verranno forniti (se si desidera), insieme agli eseguibili per crearli, in un pacchetto a parte.

NOTA IMPORTANTE: L'Editor ha bisogno dell'assegnamento Doopsi:, ma il

Player è stato costruito in modo da essere completamente rilocabile (non necessita di assegnamenti). Quando caricate il Player da una Shell dovete semplicemente far diventare la directory del Player quella corrente e poi chiamare il programma: il Player cercherà quindi la directory che avete selezionato mediante l'Editor (quella di default è la directory "Data" creata dagli archivi), dove Deve trovare i file "player.dat" e "panel.iff".

La linea di comando per caricare il Player è: Player nome_file_Doopsi. Se dimenticate di fornire il nome del file, il Player apre un file requester e vi chiede di selezionare un file.

1.8 DOOPSI-GS - Finestra Principale

```
*****  
La finestra principale.  
*****
```

La storia inizia da qui.

Prima di tutto, descriviamo i contenuti della finestra. La parte più alta è divisa in due sezioni: quella di sinistra riguarda le scene, mentre quella di destra è dedicata agli oggetti. Poi segue una riga contenente alcuni editor misti e, infine, la riga finale contiene dei gadget dedicati al caricamento e salvataggio di tutto il vostro lavoro.

Iniziamo con la sezione delle scene. Per creare una nuova scena dovete premere il bottone "Nuovo" e, dopo aver selezionato l'immagine di sfondo mediante il file requester, vi si chiede di inserire il nome della scena: questo nome è l'identificatore che Doopsi usa per indirizzare la scena, quindi siete invitati a non chiamare due scene con il medesimo nome. Dopo che lo sfondo è stato visualizzato siete pronti per lavorare su questa scena (per es.: posizionando degli oggetti su di essa). Per selezionare la scena su cui lavorare dovete usare il gadget "Lista": cliccate sul nome desiderato nella lista e poi sul bottone "Ok". Il gadget "Uccidi" serve per rimuovere la scena corrente.

Quando premete il bottone "Salva" l'editor vi chiede se volete salvare solo la scena corrente o tutte le scene create fino ad ora. Con il gadget "Carica" potete caricare entrambi i tipi di file: l'Editor riconosce automaticamente se il file che avete selezionato contiene una o più scene. Se gli oggetti agganciati alla scena che volete caricare sono già stati caricati, allora l'editor può posizionarli immediatamente sulla scena stessa; se invece caricate gli oggetti dopo che la scena è stata caricata, allora dovete posizionare tali oggetti a mano, cliccando sul gadget "R" ("R" sta per "ripristina"). Il gadget "Carica" può anche estrarre le informazioni delle scene da un file salvato con il gadget "Salva Tutto" posizionato nella fila inferiore della finestra (vedi la descrizione di tale gadget per informazioni maggiori).

I gadget rimanenti aprono altri editor, e si rimanda alle sezioni del manuale dedicate a tali editor per informazioni più specifiche. Il gadget "Attrs" serve per attivare l'editor degli Attributi delle Scene, dove potete cambiare alcune caratteristiche della vostra scena; il bottone "Spot editor" invoca lo Spot Editor per posizionare alcuni Spot sulla scena; il

gadget "Path editor" accede al Path Editor usato per creare il percorso su cui il vostro Omino dovrebbe camminare.

Ora sapete come creare una nuova scena e siete pronti per posizionarvi degli oggetti, quindi diamo uno sguardo alla sezione degli oggetti. Il gadget "Nuovo" crea un nuovo oggetto ed apre l'editor degli Attributi degli Oggetti per permettere all'utente di modificare tale oggetto (si veda la sezione dedicata all'editor degli Attributi per gli Oggetti). Il gadget "Lista" visualizza la lista di tutti gli oggetti creati fino ad ora: per lavorare su un oggetto dovete renderlo l'oggetto corrente selezionando il suo nome dalla lista. Il bottone "Uccidi" rimuove l'oggetto corrente.

Il gadget "Salva" chiede se volete salvare solo l'oggetto corrente o tutti gli oggetti creati. Il bottone "Carica" riconosce se il file che avete selezionato contiene uno o più oggetti e gli aggiunge alla lista. Questo bottone può anche estrarre i dati degli oggetti da un file salvato con il gadget "Salva Tutto".

Il gadget "Attrib" apre l'editor degli Attributi degli Oggetti (quello a cui accedete automaticamente quando create un nuovo oggetto); il bottone "Doopsi code" invoca il Doopsi Coder utilizzato per specificare come si deve comportare il Player quando l'utente agisce su un oggetto; il gadget "Anim editor" apre l'Anim Editor che vi permette di convertire un oggetto in un AnimObject costruendo un'animazione a partire dalle sue immagini.

Potete muovere tutti gli oggetti che avete creato e posizionato sulla scena (utilizzando l'Object Attributes editor) cliccando su di essi con il bottone sinistro del mouse e (tenendo premuto questo bottone) trascinandoli all'interno della scena. Cliccando su di essi con il bottone destro del mouse rendete oggetto corrente l'oggetto selezionato ed invocate l'Object Attributes editor.

La sezione successiva contiene tre bottoni: il bottone "Prefs" accede all'editor delle Preferenze dove potete modificare ad esempio alcuni nomi che il Player usa come default, o i vostri path preferiti, e così via; il "Dialog Editor" invoca il Dialog Editor utilizzato per creare dialoghi fra il vostro Omino e un altro personaggio della vostra storia; il gadget "Info" vi dice quanti oggetti e scene avete creato fino ad ora e la quantità di memoria disponibile. Infine il bottone "Prova Gioco" invoca il Player affinché possiate provare immediatamente la vostra avventura: l'Editor cerca il Player nella directory indicatagli mediante l'apposito campo situato nella finestra delle Preferenze.

Ora descriviamo l'ultima riga. Il gadget "Salva Tutto" serve per salvare tutta l'avventura: il file contiene scene, oggetti, percorsi, spot, dialoghi e preferenze, ovvero tutto quello di cui abbisogna il Player per iniziare il gioco. Il bottone "Salva Finale" differisce dal gadget "Salva Tutto" poiché il file salvato verrà crittato: l'Editor vi chiede la chiave utilizzata per crittare il file. Abbiamo aggiunto questa opzione poiché in questo modo altre persone non possano guardare nel file Doopsi per vedere come avete costruito la vostra avventura, ma possano comunque giocarci senza bisogno di conoscere la chiave. Il gadget "Carica Tutto" vi permette di caricare un file salvato con il gadget "Salva Tutto" o "Salva Finale", se volete modificarlo ulteriormente (se il file è crittato dovete conoscere

la chiave per decrittarlo prima di caricare i dati contenuti). Il bottone "Cancella Tutto" serve per cancellare tutto il vostro precedente lavoro e ricominciare da capo. Il bottone "Info su Doopsi" visualizza un piccolo messaggio che dovrete leggere.

I gadget.

Sezione delle scene:

Nuovo	Crea una nuova scena: vi si chiede di selezionare l'immagine di sfondo e di scrivere il nome della scena.
Carica	Carica un file contenente una o più scene.
Salva	Salva un file contenente dati per le scene: vi si chiede se volete salvare solo la scena corrente o tutte le scene create fino ad ora.
Lista	Lista tutte le scene create e vi permette di muovervi fra le scene selezionando quella corrente.
Uccidi	Cancella la scena corrente.
Attrs	Apri lo Scene Attributes editor.
Spot editor	Invoca lo Spot editor.
Path editor	Accede al Path editor.
R	Ridisegna lo schermo.

Sezione degli oggetti:

Nuovo	Crea un nuovo oggetto ed apre l'Object Attributes editor per poterlo modificare.
Carica	Carica un file contenente uno o più oggetti.
Salva	Salva un file contenente dati per gli oggetti: vi si chiede se volete salvare solo l'oggetto corrente o tutti gli oggetti creati.
Lista	Lista tutti gli oggetti: potete muovervi fra di essi selezionando quello corrente.
Uccidi	Cancella l'oggetto corrente.
Attrib	Apri l'Object Attributes editor.
Doopsi code	Invoca il Doopsi Coder.
Anim editor	Accede all'Anim editor.

Altri gadget:

Prefs	Apri l'editor delle Preferenze.
Dialog editor	Accedi al Dialog editor.
Prova Gioco	Invoca il Player.
Info Progetto	Visualizza alcune informazioni sul vostro lavoro.
Carica Tutto	Carica un file contenente tutti i dati del gioco (se il file è crittato vi viene chiesta la chiave per decrittarlo).
Salva Tutto	Salva un file contenente tutti i dati del gioco, cioè il file utilizzato dal Player.
Salva Finale	Salva un file crittato contenente tutti i dati del gioco: l'Editor vi chiede la chiave per crittare il file.
Canc. Tutto	Elimina tutto il vostro precedente lavoro.
Info su Doopsi	Alcune note a proposito di noi e di Doopsi.

1.9 DOOPSI-GS - Note

```
*****
Alcune note su come il Player trova l'Omino.
*****
```

Questo problema sorge in quanto l'Omino è un AnimObject (in realtà quattro oggetti: uno per ogni direzione) come tutti gli altri AnimObject che potete agganciare al vostro gioco. Quindi il Player deve sapere quali oggetti rappresentano il protagonista della storia. Il nome interno dell'Omino è composto da una radice (quella di default è "Omino") ed una estensione che indica la direzione (ad es.: A per alto, G per giù, ecc.): quindi l'AnimObject contenente l'animazione che rappresenta l'Omino che cammina verso destra potrebbe essere chiamata (sempre nome interno!) "OminoD", e così via. Avete bisogno anche di altri quattro AnimObject rappresentanti l'Omino che parla nelle quattro direzioni: il nome di tali AnimObject è costruito a partire dalla radice, aggiungendo ad essa una prima estensione (quella di default è "Parla") e la seconda estensione è la direzione, come prima. Quindi un nome tipico per l'AnimObject rappresentante l'Omino che parla verso sinistra è "OminoParlaS".

Potete scegliere la radice e le estensioni che preferite scrivendole nell'editor delle preferenze.

Un'ultima nota a proposito di estensioni. Il Doopsi Coder contiene istruzioni utilizzate per muovere (e/o animare) oggetti sulla scena durante il gioco: vi si chiede di fornire il nome dell'AnimObject contenente l'animazione desiderata. E qui arriva la parte interessante: se tale nome termina con una delle quattro estensioni indicanti una direzione, allora il Player è autorizzato a scegliere la direzione che gli serve. Fa ciò rimuovendo l'estensione e rimpiazzandola con quella indicante la direzione

giusta, poi cerca un AnimObject con il nome risultante dall'operazione precedente e, se lo trova, lo usa. Per esempio: se il nome che avete selezionato come argomento per l'istruzione SetMoveAnim è "VolaD" e la mosca si deve muovere verso l'alto, allora il Player cerca un AnimObject chiamato "VolaA". In altre parole: l'estensione che voi fornite al momento della programmazione è irrilevante perché verrà rimpiazzata prima di essere utilizzata, ma essa dice al Player di stare attento alla direzione. Per sapere quali istruzioni hanno questa opzione consultate l'Appendice A.

Osservazioni generali che dovrete leggere:

L'Editor produce un file contenente la struttura della vostra avventura, ma non la grafica o i suoni che avete agganciato al gioco poiché non vogliamo mescolare tutta l'avventura in un solo file enorme; quindi nel file generato vi sono solo i percorsi dei file agganciati. Dovete ricordarvi di questo particolare se volete distribuire il vostro gioco agli amici: per esempio, potete far diventare relativi i percorsi utilizzando un'assegnamento.

1.10 DOOPSI-GS - Path Editor

```
*****  
Path Editor.  
*****
```

Introduzione.

Il Path Editor si occupa di aiutare il creatore dell'avventura a sviluppare i percorsi nei quali si muoveranno i vari personaggi. La creazione dei percorsi è forse una delle cose più complesse e richiede una certa attenzione. A ben vedere, il percorso che verrà creato su di una scena vincolerà i movimenti dei personaggi durante il gioco. Questo deve necessariamente portare ad una pianificazione ponderata per la creazione del percorso.

Teoria dei path.

Il percorso, o path, è strutturato in nodi, delle unità di sosta. Ogni nodo può avere delle caratteristiche particolari che permettono un maggiore controllo del movimento del personaggio sulla scena. Di queste caratteristiche parleremo più avanti. Ogni nodo ha dei "rami" (branch) al massimo sette per ogni nodo, che permettono di collegarsi ad altri nodi.

Il path determina, attraverso i nodi, la possibilità di movimento del personaggio sulla scena. Un percorso è quindi formato da nodi, uniti tra loro da collegamenti (branch), in modo da formare una specie di "rete". I movimenti del personaggio dell'avventura sulla scena avverrà appunto su questa rete, come se i nodi con le loro diramazioni fossero una specie di "binario invisibile" sul quale il personaggio si muove. Non è possibile, in alcun modo, muovere il personaggio in zone della scena non raggiungibili da un nodo. Se si chiede al personaggio di muoversi in un luogo esterno

alla diramazione del percorso, verrà raggiunto il punto ritenuto più vicino a quello richiesto, ma non si andrà oltre. Questo aspetto dei path può sembrare una limitazione, ma è in realtà una delle caratteristiche che assicurano una grandissima flessibilità a livello di game-design: per esempio, è possibile creare una scena che racchiude più percorsi e "vietare" l'accesso a certe aree a meno che non si siano verificate determinate condizioni.

Con un po' di pratica, riuscirete sicuramente a creare dei percorsi complessi ed articolati. Il Path Editor vi assisterà costantemente e con affidabilità in questo compito. Buon lavoro.

NOTA: I seguenti tutorial presuppongono che voi conosciate già altre caratteristiche proprie di Doopsi, quali la creazione di nuove scene, la creazione di animazioni tramite AnimWorkshop e la gestione degli oggetti tramite l'editor di Attributi dell'Oggetto.

Path Tutorial 1

Path Tutorial 2

Descrizione Generale del Path Editor

1.11 DOOPSI-GS - Path Tutorial 1

Tutorial 1 - Un Semplice Percorso

In questo tutorial impareremo a disegnare un piccolo percorso di quattro nodi e ad unire questi nodi tra loro con delle semplici diramazioni. Scopo di questo tutorial è di impraticarvi con i comandi di base del Path Editor. Imparerete ad aggiungere e cancellare nodi, ad aggiungere collegamenti e a forzare il ridisegnamento di tutta la scena (necessario in alcuni casi). Nel prossimo tutorial costruiremo qualcosa di più complesso. Eseguite questa esercitazione anche più volte, fino a quando non sarete certi di avere completa padronanza dei comandi.

Per eseguire questo tutorial dovrete aver creato prima una scena con l'immagine "tutorial_room1".

Premete Path Editor nel pannello principale di Doopsi.

Non fatevi spaventare dal numero di pulsanti che ci sono, col tempo ne avrete piena padronanza.

Prima di tutto, è consigliabile sistemare tutti i nodi che si desiderano sullo schermo. Naturalmente è possibile aggiungerne anche in seguito, ma in questo modo sarete certi di coprire almeno tutti i punti importanti.

Per aggiungere un nodo sulla scena è sufficiente eseguire questi passaggi:

- Premete "Agg. Nodi"
Da questo momento in poi, fino a quando non decideremo di smettere, ogni click del mouse sulla scena darà origine alla creazione dei nodi.
 - Create quattro nodi in modo che formino una specie di quadrato.
-

- Premete "Agg. Coll." per aggiungere i collegamenti.
Da questo momento in poi, fino a quando non decideremo interrompere l'operazione, potremo aggiungere collegamenti tra i nodi.
- Premete sul primo nodo (il nodo cambierà di colore)
- Premete sul secondo nodo (il nodo cambierà di colore e una linea unirà i due nodi)

Abbiamo appena creato il primo collegamento. Per creare il secondo, procediamo in questo ordine:

- Premete sul secondo nodo (il nodo cambierà colore)
- Premete sul terzo nodo (il nodo cambierà colore e una linea unirà i due nodi)

Proseguite in questo modo fino a quando non avrete creato un vero "quadrato".

Adesso che avete creato questi nodi, potete testare lo spostamento che effettuerà il personaggio durante l'avventura:

- premete il bottone "Prova Movim." e selezionate il nodo di partenza (per esempio il primo). Il nodo si illuminerà.
- Selezionate il secondo nodo (ad esempio il terzo). Il nodo si illuminerà.

Il programma cercherà di calcolare un percorso tra i due nodi selezionati e lo visualizzerà, passo passo, illuminando i nodi. Se avete selezionato il primo ed il terzo e li avete collegati come descritto nel nostro esempio, dovreste vedere illuminarsi i nodi uno, due e tre. Naturalmente questo è un esempio semplice, ma la possibilità di provare il path si rivela molto utile in scene complesse.

Adesso create un nuovo nodo dove volete e collegategli altri nodi, seguendo i procedimenti appena descritti. Riprovate ad utilizzare "Prova Movim.".

Prima di terminare il tutorial, proviamo a rimuovere un nodo dalla scena:

- Premete "Uccidi Nodo"
- Selezionate il nodo che intendete cancellare.
- Il nodo verrà cancellato.

A questo punto, è possibile che della grafica sulla scena sia rovinata e che il nodo appena cancellato compaia ancora. Questo non è un errore di Doopsi, ma è necessario eseguire un aggiornamento della grafica (refresh). Premete "Aggiorna" per aggiornare la grafica, vedrete che adesso il nodo cancellato non comaprà più.

1.12 DOOPSI-GS - Path Tutorial 2

Tutorial 2 - La programmazione dei nodi

I percorsi sono strumenti potentissimi. Per utilizzarli al meglio è

necessario provare e riprovare. Quello che cercheremo di spiegarvi adesso è la "programmazione" di un nodo.

Un nodo può essere programmato per le seguenti attività:

- Modificare l'animazione del personaggio
- Modificare la priorità uomo/oggetti
- Determinare il cambio di scena
- Controllare se un nodo è percorribile (controllo di condizione)

Tutte queste operazioni sono complesse da realizzare e richiedono uno studio accurato della generazione del percorso.

Vediamo adesso come è possibile programmare un nodo e quali sono le condizioni da tenere in considerazione nella progettazione di un percorso.

Per programmare un nodo, è sufficiente premere il bottone "Codice" nel Path Editor e poi selezionare il nodo che si intende modificare.

- Modificare l'animazione del personaggio

Premere il bottone "Setta Anim.", apparirà un Lister con tutti gli oggetti (animati e non) presenti nell'avventura. Selezionate quello che vi riguarda (Deve essere un AnimObject).

- Modificare la priorità uomo/oggetti

Premete il bottone "Cambia Pri", verranno alternate le scritte "Uomo Su Oggetti" o "Oggetti Su Uomo".

- Determinare il cambio di scena

Premere su "Set Start", apparirà un Lister contenente tutte le scene create. Selezionate quella nella quale desiderate spostarvi quando il personaggio passerà sopra al nodo.

- Controllare se un nodo è percorribile (controllo di condizione)

Premete "Set Condizione", apparirà un Lister con tutti i nomi degli oggetti presenti nell'avventura. Selezionato l'oggetto, comparirà uno status requester per determinare la condizione che andrà controllata su quel determinato nodo.

Facciamo un piccolissimo esempio di utilizzo:

Immaginate di avere i seguenti nodi:

1 ---- 2 ---- 3 ---- 4 ---- 5

Noi vogliamo che il nodo 5 ci faccia saltare alla scena "Ufficio", ma solo quando l'oggetto "porta" è aperta (supponiamo che lo status sia a 1 quando la porta è aperta). Dobbiamo fare in modo che il protagonista NON possa

raggiungere il nodo n.5 se non quando la porta è aperta.

Iniziamo a programmare il nodo 5. Premete "Codice" e poi sul nodo 5, e cliccate su "Set Start", selezionando la scena "Ufficio": da ora in poi tutte le volte che il personaggio arriverà sul nodo n.5, passerà alla scena "Ufficio".

Ora programmate il nodo n.4: premete "Codice" poi selezionate il nodo e cliccate su "Set Condizione". Premete sull'oggetto "Porta" e inserite come valore di status 1. Fatto! Adesso, se la porta è chiusa, il nodo n.5 non sarà raggiungibile.

1.13 DOOPSI-GS - Descrizione Generale del Path Editor

Descrizione generale.

Probabilmente questo è l'editor con più bottoni, ma questi gadget sono raccolti in gruppi. Esamineremo un gruppo alla volta.

Il primo gruppo che mi accingo a descrivere è quello situato nell'angolo in alto a sinistra della finestra. Il bottone "Agg. Nodi" attiva l'aggiunta dei nodi, cioè, dopo aver premuto questo gadget potete aggiungere quanti nodi volete fino a quando siete soddisfatti: dovete solo cliccare sulla scena nel punto dove volete un nodo. Per uscire dall'aggiunta dei nodi dovete premere ancora il gadget "Agg. Nodi" (o un altro gadget del medesimo gruppo). Il bottone "Uccidi Nodo" vi chiede di selezionare il nodo da uccidere, dopo di che l'Editor lo rimuoverà dalla scena: il nodo "morto" lascerà un quadrato nero sulla scena, così, se volete restaurare la vostra grafica, dovete premere il bottone "Aggiorna". Il bottone "Aggiorna" ridisegna tutta la grafica contenuta nella scena corrente. Il bottone "Agg. Coll." serve per attivare l'aggiunta dei collegamenti: per aggiungere un collegamento fra due nodi premete prima sul nodo iniziale e poi su quello finale. Potete ripetere questi passaggi fino a quando non avrete finito di collegare tutti i vostri nodi; quindi premete il bottone "Agg. Coll." ancora una volta per uscire dall'aggiunta dei collegamenti. Il gadget "Codice" vi chiede di selezionare il nodo da programmare; per terminare la sessione di programmazione premete ancora sul bottone "Codice": l'Editor vi chiederà di selezionare un'altro nodo da programmare, ma potete cliccare nel campo del "Messaggio" per farlo smettere definitivamente.

La programmazione dei nodi ci porta direttamente al gruppo intitolato "Modifica Settaggi". Il bottone "Setta Anim." è utilizzato per cambiare le animazioni dell'Omino quando l'Omino passa su quel nodo: l'Editor apre un lister con tutti i nomi degli oggetti creati ed aspetta che l'utente selezioni un nome dalla lista. A questo punto il testo "Default" alla sinistra del gadget viene rimpiazzato con il nome scelto. Da questo nodo in poi il Player userà la radice del nome scelto come radice per costruire tutti i nomi delle animazioni dell'Omino (se questo non vi dice niente allora dovrete leggere le "Alcune note su come il Player trova l'Omino" situate appena dopo la sezione della Finestra Principale di questo manuale). Il gadget "Cambia Pri." alterna la priorità fra "Uomo Su Oggetti" a "Oggetti Su Uomo" e viceversa: la prima condizione significa che l'Omino apparirà davanti a tutti gli oggetti mentre cammina verso quel nodo, la seconda significa giusto il contrario (potete trovare altre note a

proposito di come questo funzioni nella sezione dedicata al Player del manuale). Il bottone "Set Start" apre un lister con i nomi di tutte le scene create fino ad ora: il nome della scena selezionata apparirà alla sinistra del gadget nel campo "Start". Se questo campo non è vuoto il Player, quando l'Omino calpesta quel nodo, cerca una scena avente il nome scelto; se la scena viene trovata allora il Player abbandona la scena precedente ed entra in quella nuova. Nella nuova scena l'Omino verrà posizionato sul nodo contenente (nel campo "Start") il nome della scena da cui l'Omino proviene. (E' come se il nodo "finale" della prima scena punti verso il nodo "iniziale" della seconda, e viceversa.)

Il prossimo gruppo è quello chiamato "Condizioni". Il bottone "Set Condizione" permette all'utente di arrestare l'Omino se una determinata condizione non è verificata: la condizione è del tipo "Se lo stato dell'oggetto è uguale a". Il bottone apre un lister con i nomi degli oggetti; dopo aver selezionato un oggetto dovete inserire il valore dello stato che volete controllare. Se il campo di Stato dell'oggetto selezionato non contiene il valore scelto quando l'Omino cammina su quel nodo, allora all'Omino non è permesso oltrepassare tale nodo. Il gadget "Cancella Condiz." cancella i campi di condizione del nodo.

Il bottone "Prova Movim." vi permette di vedere come l'Omino camminerà sul vostro percorso: vi si chiede di selezionare il nodo di partenza ed il nodo di arrivo; a questo punto l'Editor illuminerà i nodi su cui l'Omino passerà mentre va dal nodo iniziale a quello finale. Il bottone "Controlla Nodi" controlla se tutti i nodi sono correttamente collegati e rimuove i nodi isolati.

Ci mancano solo tre gadget. Il bottone "Nuovo Path" cancella completamente il path attuale. Il gadget "Salva Path" salva il path corrente in un file, e il gadget "Carica Path" carica un file salvato con il precedente bottone.

Alcuni suggerimenti utili:

Premete sempre il bottone "Controlla Nodi" prima di salvare tutto il vostro lavoro, poiché in questo modo vi eviterete sgradite sorprese.

Quando posizionate i nodi non dimenticatevi delle dimensioni dell'Omino: l'hot spot dell'Omino è posizionato nell'angolo in basso a sinistra dell'immagine dell'Omino, quindi l'Omino si ferma sempre alla destra di un nodo. Questo implica che l'Omino può raggiungere il bordo destro (o il bordo superiore) della scena prima di raggiungere il nodo di destinazione se posizionate tale nodo troppo vicino al bordo della scena: quando l'Omino raggiunge il bordo della scena si ferma, così il vostro nodo non verrà mai raggiunto.

Quando stabilite la priorità dell'Omino sugli oggetti tenete presente le dimensioni dell'Omino: se un nodo dice "Oggetti Su Uomo" ed il seguente dice "Uomo Su Oggetti" e l'Omino, andando dal primo al secondo nodo, è ancora parzialmente coperto da un oggetto, esso può venire improvvisamente posizionato davanti all'oggetto che lo copre. Di conseguenza vi capiterà spesso di dover scegliere la priorità "Oggetti su Uomo" anche per un nodo che non è coperto da alcun oggetto per essere sicuri che l'Omino sia completamente scoperto prima di cambiare la sua priorità.

I gadget.

Agg. Nodi	Attiva l'aggiunta dei nodi: cliccate sulla scena per posizionare quanti nodi volete, e poi cliccate su questo gadget per uscire dall'aggiunta.
Uccidi Nodo	Chiede di selezionare un nodo e poi lo cancella. Cliccate ancora su questo bottone per annullare l'operazione.
Agg. Coll.	Attiva l'aggiunta dei collegamenti: cliccate sul nodo iniziale e su quello finale fino a quando non avrete posizionato tutti i collegamenti. Premete ancora questo gadget per uscire dall'aggiunta.
Codice	Attiva la sessione di programmazione. Quando avete finito di cambiare gli attributi del nodo premete ancora questo bottone.
Aggiorna	Ridisegna tutta la grafica della scena.
Setta Anim.	Cambia il campo "Anim" del nodo.
Cambia Pri	Cambia la priorità dell'Omino rispetto agli oggetti.
Set Start	Seleziona la scena in cui entrare quando l'Omino cammina su questo nodo.
Set Condizione	Stabilisce la condizione che deve essere verificata affinché l'Omino possa oltrepassare il nodo.
Cancella Condiz.	Cancella il campi di condizione di un nodo.
Nuovo Path	Cancella il path corrente.
Salva Path	Salva il path corrente in un file.
Carica Path	Carica un path da un file.

1.14 DOOPSI-GS - Spot Editor

```
*****  
Spot Editor.  
*****
```

Questo vi permette di posizionare degli "Spot" sulla scena: essi sono semplicemente delle piccole crocette con due coordinate ed un nome. Voi usate questo nome per designare un particolare spot e Doopsi userà le sue coordinate. Gli spot vengono usati assieme, ad esempio, all'istruzione

MoveObject per indicare il posto dove l'oggetto dovrà recarsi.

Spot Tutorial

Descrizione Generale dello Spot Editor

1.15 DOOPSI-GS - Spot Tutorial 1

Eccovi un TUTORIAL:

- Create una nuova scena (utilizzando il bottone "New" nella sezione dedicata alle scene della finestra principale dell'editor).
- Attivate lo Spot editor.
- Selezionate il gadget "Aggiungi Spot": apparirà il messaggio "Aggiungi gli spot".
- Cliccate sulla scena dove volete mettere uno spot: potete ripetere questa operazione fino a quando avrete posizionato tutti i vostri spot.
- Selezionate ancora il bottone "Aggiungi Spot" per disattivare il modo di aggiunta degli spot.
- Cliccate sulla scena per selezionare uno spot: apparirà il messaggio "Informazioni".
- Potete modificare il nome dello spot corrente semplicemente scrivendolo nello string gadget "Nome Spot" e premendo Return.
- Uscite dallo Spot editor.

1.16 DOOPSI-GS - Descrizione Generale dello Spot Editor

Descrizione generale.

Per aggiungere alcuni spot dovete cliccare sul bottone "Aggiungi Spot": quindi cliccate sulla scena ovunque vogliate posizionare uno spot. Poiché uno spot senza un nome non ha senso lo Spot Editor attribuisce ai vostri spot un nome di default, nome che potete cambiare in ogni momento semplicemente riscrivendolo nel gadget stringa chiamato "Nome Spot" e premendo Return. D'ora in poi potete aggiungere spot a vostro piacimento. Quando siete soddisfatti, potete interrompere l'aggiunta di spot premendo ancora il bottone sinistro del mouse sul gadget "Aggiungi Spot" o su quello chiamato "Elimina Spot" o, infine, su qualunque parte della finestra dell'editor. Per eliminare uno spot premete una volta il gadget "Elimina Spot": l'editor vi chiederà di scegliere lo spot da uccidere, e quindi non vi rimane che cliccare su tale spot. Questo è tutto. A proposito, dopo aver ucciso uno spot (o dopo aver deciso di non ucciderlo cliccando ancora nella finestra dell'editor) è necessario rilescezionare il gadget "Elimina Spot" per ucciderne un'altro, e così via.

L'azione di default (quando non state aggiungendo o eliminando spot) è chiamata "Informazioni": siete liberi di cliccare su qualunque spot per illuminarlo e visualizzare il suo nome nello string gadget. Come ho detto in precedenza, questo nome può essere modificato ogni volta che viene visualizzato.

Manca ancora un gadget: il bottone "Aggiorna". Dopo aver posizionato (ed eliminato) molti spot la vostra scena potrebbe apparire un po' sottosopra: cliccate su questo gadget e Doopsi pulirà la stanza per voi.

I gadget:

Aggiungi Spot	Attiva (e disattiva) l'aggiunta degli spot.
Elimina Spot	Cancella uno spot.
Nome Spot	Modifica il nome dello spot.
Aggiorna	Ridisegna la scena.

1.17 DOOPSI-GS - Attributi della Scena

```
*****
Screen Attributes editor.
*****
```

Questo editor vi permette di cambiare il nome della scena corrente, di caricare uno sfondo diverso senza rimuovere oggetti, percorso e spot che avete già agganciato a quella scena, ed altre cose utili...

L'ultima riga in basso dice qualcosa del genere: "Usa Azione Apri di Oggetto pippo". In altre parole il codice Doopsi contenuto nell'azione specificata dell'oggetto selezionato verrà eseguito tutte le volte che entrerete nella scena durante il gioco. Per disattivare questa opzione dovete cancellare il nome dell'oggetto nello string gadget. Un suggerimento utile: se volete che il codice sia eseguito solo la prima volta in cui entrate nella scena (o in occasioni speciali) potete sfruttare il campo di stato dell'oggetto: inserite in questo campo un valore particolare ed iniziate il codice Doopsi con l'istruzione "IfStatus".

I gadget:

Cambia Sfondo	Carica un nuovo sfondo per la scena. Se almeno un nodo del percorso esistente o uno spot cadono al di fuori della bitmap allora l'editor vi avviserà senza, però, rimuovere l'oggetto incriminato.
Nome Scena	Cambia il nome interno della scena.
Nome Musica	Potete inserire il nome di un modulo che verrà suonato ogni volta che entrerete nella scena (durante il gioco, ovviamente) e verrà interrotto quando uscirete dalla scena stessa. Potete cliccare sul bottone per aprire un

file requester e selezionare il modulo, oppure potete scriverne direttamente il nome nel gadget stringa.

Usa Azione Questo cycle gadget vi permette di scegliere l'azione da cui leggere il codice Doopsi che è eseguito quando entrate nella scena durante il gioco.

di Oggetto Seleziona l'oggetto da cui leggere il codice Doopsi specificato dal gadget "Usa Azione". Questo bottone apre una lista con i nomi di tutti gli oggetti. Potete anche inserire il nome dell'oggetto a mano utilizzando il gadget stringa.

1.18 DOOPSI-GS - Attributi degli Oggetti

```
*****  
L'Editor degli Attributi di un Oggetto.  
*****
```

Questo editor vi permette di specificare le caratteristiche di un oggetto e di posizionarlo sulla scena.

Tutorial 1
Tutorial 2

Descrizione Generale degli Attributi di un Oggetto

1.19 DOOPSI-GS - Attributi di un Oggetto: Tutorial 1

Un primo tutorial: "il foglietto".

- Suppongo che abbiate già caricato una scena per potervi posizionare degli oggetti. Ora create un nuovo oggetto premendo il bottone "Nuovo" nella sezione dedicata agli oggetti della finestra principale.

- Attivate l'editor degli Attributi.

- Nel gadget "Nome" scrivete "il foglietto".

- Cliccate sul gadget "Nome Interno" e scrivete "foglietto".

- Cliccate sul cycle gadget per selezionare "Grafica" come tipo di oggetto, in quanto il nostro oggetto dovrà essere disegnato sulla scena.

- Premete il bottone "Agg." e, usando il file requester, scegliere l'immagine "paper" nella directory Tutorial. Ora l'immagine è agganciata all'oggetto.

- Infine premete il bottone "Pos" per posizionare l'oggetto sulla scena.

- Potete muovere l'oggetto nella sua posizione finale selezionandolo con il mouse e trascinandolo all'interno della scena.

1.20 DOOPSI-GS - Attributi di un Oggetto: Tutorial 2

Un secondo tutorial: "una zona vuota".

- Create un nuovo oggetto come nel precedente tutorial.
- Scrivete il suo nome e (più importante) il suo nome interno.
- Cliccate sul cycle gadget per selezionare "Vuota" come tipo di oggetto: il nostro oggetto sarà una cornice vuota (che non verrà disegnata sullo schermo durante il gioco).
- Premete il bottone "Setta Zona": un rettangolo apparirà sulla scena.
- Trascinate il rettangolo nella scena con il mouse e scalatelo utilizzando il gadget posizionato nell'angolo in basso a destra del rettangolo stesso.
- Quando avete finito premete il gadget "Pos" per agganciare questo oggetto alla scena.

1.21 DOOPSI-GS - Descrizione Generale degli Attributi di un Oggetto

Descrizione generale.

Potete attivare questo editor cliccando sul bottone "Attrib" nella sezione dedicata agli oggetti della finestra principale o cliccando il tasto destro del mouse su un oggetto già posizionato sullo schermo.

I primi campi che dovete riempire sono quelli indicati con "Nome" e "Nome Interno": il nome di un oggetto è la stringa che appare sulla linea di testo della console del Player quando il puntatore è su tale oggetto; il nome interno è l'identificatore che Doopsi usa per indirizzare un oggetto: per questa ragione siete invitati a non usare più volte lo stesso nome.

Dopodiché concentriamoci sulla zona delle immagini. Per aggiungere alcune immagini al vostro oggetto dovete cliccare sul bottone "Agg.": l'editor aprirà un file requester e, dopo che avrete selezionato un file valido, visualizzerà il nome dell'immagine (completa di percorso) nella lista. Per eliminare un'immagine non dovete far altro che cliccare su tale immagine e poi premere il bottone "Uccidi". Poi dovete dire all'editor quale immagine usare per disegnare l'oggetto sulla scena; per far ciò selezionate una immagine nella lista e premete il gadget "Shp Def.": apparirà un numero che mostra qual è l'immagine corrente. Una procedura simile è necessaria per dire al Player qual è l'immagine di inventario: cliccate su un'immagine e poi sul bottone "Shp Inv.". A proposito: le immagini di inventario devono avere dimensioni di 32*32 pixel, altrimenti verranno tagliate prima di essere visualizzate dal Player. Vi è ancora un gadget da descrivere: il cycle gadget è necessario per scegliere se si vuole un oggetto "Grafico" od uno "Vuoto". Il primo ha un'immagine corrente che viene disegnata sulla scena, mentre il secondo è solo un rettangolo vuoto utilizzato per incorniciare qualcosa appartenente allo sfondo. Per posizionare e ridimensionare tale rettangolo dovete cliccare sul bottone "Setta Zona": una Zona Vuota verrà disegnata sulla scena, zona che potete trascinare col mouse cliccando su di essa; per riscalarla premete

semplicemente sul quadrato pieno nell'angolo in basso a destra della zona stessa. E' importante notare che anche oggetti Vuoti possono avere alcune immagini agganciate: potete usare queste immagini con i comandi del Player che riguardano l'inventario. Ciascun oggetto ha spazio per più di una immagine e più di una immagine di inventario poiché durante il gioco potete cambiare l'immagine corrente dell'oggetto o la sua immagine di inventario utilizzando le istruzioni del Player relative alle immagini.

Nel gadget "Stato" potete inserire il valore di partenza dello stato dell'oggetto. Il campo di stato è un valore intero privo di un significato particolare, dimodoché lo potete usare per qualunque scopo vi aggradi, ad es.: potete memorizzare qui lo stato Aperto/Chiuso di una porta scrivendo 0 per Chiuso e 1 per Aperto, e potete cambiare questo stato durante il gioco usando ChangeStatus ed altre istruzioni del Player.

Uscendo da questo editor ricordate di premere sempre il bottone "Pos" se volete agganciare l'oggetto alla scena, anche se l'avete già fatto in precedenza: questo perché il gadget "Ok" accetta le modifiche apportate all'oggetto ma sgancia sempre tale oggetto dalla scena. Non c'è bisogno di spiegare cosa fa il gadget "Cancella".

I gadget:

Nome	Nome dell'oggetto.
Nome Interno	Nome interno dell'oggetto.
Shp Def.	Questo intero punta all'immagine utilizzata per disegnare l'oggetto sulla scena. Non ha significato alcuno per una Zona Vuota.
Shp Inv.	Questo intero punta all'immagine utilizzata dal Player per immagazzinare l'oggetto nell'inventario (le dimensioni di questa immagine devono essere di 32*32 pixel).
Vuota/Grafica	Seleziona il tipo di oggetto.
Agg.	Aggancia un'immagine all'oggetto e aggiunge il suo nome alla lista.
Uccidi	Rimuove un'immagine dalla lista.
Stato	Modifica il valore corrente dello stato.
Setta Zona	Crea il rettangolo di una Zona Vuota.
Pos	Esce agganciando l'oggetto alla scena.
Ok	Esce senza agganciare l'oggetto alla scena (se l'oggetto è già agganciato, allora questo gadget lo sgancia).
Cancella	Esce ripristinando le condizioni iniziali.

1.22 DOOPSI-GS - DOOPSI Coder

```
*****  
Doopsi Coder.  
*****
```

Introduzione.

Doopsi Coder è l'editor che vi permette di "programmare" gli oggetti a reagire alle azioni dell'utente. Per il player di DOOPSI, gli oggetti non sono altro che complesse strutture di dati senza però alcun vero significato. Compito del Coder è di permettere un coordinamento tra le azioni possibili dal giocatore e il comportamento di ogni oggetto. Il Coder è forse la parte più importante nell'editing di una avventura, in quanto permette effettivamente al creatore della storia di dare una forma "reale" agli oggetti in scena e di permettere al giocatore di interagire con gli oggetti stessi.

La teoria che è alla base del Doopsi Coder è molto semplice: ad ogni azione permessa all'utente potrebbe corrispondere una reazione da parte dell'oggetto interessato. Abbiamo detto "potrebbe" perché non è necessario che ogni oggetto sia in grado di rispondere a tutte le azioni che l'utente desidererebbe effettuare. Se ad esempio il giocatore volesse parlare con la porta, quest'ultima non è tenuta a rispondere (a meno che non si tratti di una porta magica...)

La programmazione degli oggetti in Doopsi avviene all'interno del Coder, che garantisce un ambiente semplice e rapido per la loro organizzazione. Non preoccupatevi se non avete dimestichezza con la programmazione: non dovrete scrivere neppure una linea di codice, in quanto il Coder rende questo compito accessibile a chiunque: tutto quello che dovrete fare è di avere bene in mente quello che desiderate ottenere da una determinata azione su di uno specifico oggetto.

Nozioni generali di OOP in Doopsi.

Doopsi significa "Dynamic Object Oriented Programming System Interface", Interfaccia di Sistema per la Programmazione Dinamica Orientata agli Oggetti. I programmatori smaliziati noteranno che in Doopsi non è presente un vero e proprio interprete OOP, bensì è Doopsi stesso che si occupa di gestire l'interfacciamento oggetto-codice. Per questo la programmazione OOP in Doopsi è Dinamica: gli oggetti possono mutare forma e caratteristiche "al volo". Gli oggetti sono visti come delle "entità", un insieme di dati e caratteristiche che li rendono unici e riconoscibili. Tutti i dati contenuti all'interno di un oggetto sono protetti e non direttamente accessibili. L'accesso è garantito solo tramite le istruzioni del Doopsi Language che si occupa di "smistare" i comandi e di accedere correttamente ai dati desiderati. Molti comandi chiedono come primo parametro il nome interno dell'oggetto sul quale dovranno agire. Questo significa che il codice di un oggetto può modificare le caratteristiche di un altro oggetto del gioco. Ogni oggetto possiede uno "Status", un valore che può essere modificato con del codice Doopsi e che rappresenta quello che il programmatore desidera: il valore di Status è arbitrario e non è legato all'oggetto, dipende al programmatore definire come utilizzare Status.

I comandi del Doopsi-Language sono elencati completamente nell'Appendice A.

NOTA: i seguenti tutorial presuppongono che voi abbiate padronanza di altre parti di Doopsi, come l'Editor degli Attributi dell'Oggetto. Vedere i Tutorial del capitolo "Creazione di Oggetti" se non si hanno ben chiari i principi della creazione degli oggetti.

Tutorial 1
Tutorial 2

Descrizione Generale di DOOPSI Coder

1.23 DOOPSI-GS - Coder Tutorial 1

Tutorial 1 - Una porta che si apre e si chiude

Questo esempio presuppone che abbiate già creato un oggetto chiamato internamente "door" che possiede due immagini, la prima raffigurante la porta chiusa ("door_closed") e la seconda raffigurante la porta aperta ("door_open").

- Dall'editor principale, selezionate l'oggetto "door" come corrente e premete il bottone Doopsi Coder, sarete così pronti a programmare l'oggetto.

- Premete il bottone "Apri" per programmare la reazione dell'oggetto "door" all'azione "Apri". Eseguite questi passi:

- Comando "IfStatus"
- Selezionate l'oggetto "door"
- Inserite 0 come valore
(Nella finestra del Coder comparirà il comando: IfStatus SELF, 00)
- Comando "ChangeStatus"
- Selezionate l'oggetto "door"
- Inserite 1 come valore
(Nella finestra del Coder comparirà il comando: ChangeStatus SELF, 01)
- Comando "ChangeShape"
- Selezionate l'oggetto "door"
- Selezionate l'immagine raffigurante la porta aperta (ad es. "door_open")
(Nella finestra del Coder comparirà il comando: ChangeShape SELF, 01)
- Comando "Else"
- Comando "ShowText"
- Inserite "La porta è già aperta"
(Nella finestra del Coder comparirà il comando: ShowText La portà è già aperta)
- Comando "EndIf"

Al termine della sequenza appenna effettuata, nella finestra del Coder comparirà questo codice:

```
IfStatus SELF, 00
ChangeStatus SELF, 01
ChangeShape SELF, 01
```

```
Else
ShowText La porta è già aperta
EndIf
```

Questo codice significa (spiegato riga per riga):

```
Se lo status dell'oggetto stesso (SELF, la porta) è 0
Cambia lo status dell'oggetto stesso (SELF, la porta) a 1
Cambia l'immagine dell'oggetto stesso (sempre la porta) a 1 ("door_open")
Altrimenti (Else)
Mostra il Messaggio "La porta è già aperta"
Fine del controllo.
```

Scriviamo adesso il codice necessario per l'azione "Chiudi". Premete il bottone "Chiudi" per programmare la risposta dell'oggetto porta all'azione di chiusura. Eseguite questi passi:

- Comando "IfStatus"
- Selezionate l'oggetto "door"
- Inserite il valore 1
- Comando "ChangeStatus"
- Selezionate l'oggetto "door"
- Inserite il valore 0
- Comando "ChangeShape"
- Selezionate l'oggetto "door"
- Selezionate l'immagine "door_closed"
- Comando "Else"
- Comando "ShowText"
- Inserite "La porta è già chiusa"
- Comando "EndIf"

Comparirà il seguente codice:

```
IfStatus SELF, 01
ChangeStatus SELF, 00
ChangeShape SELF, 00
Else
ShowText La porta è già chiusa
EndIf
```

Che "tradotto" significa:

```
Se lo Status dell'oggetto stesso (SELF, porta) è a 1
Cambia lo Status dell'oggetto stesso (SELF, la porta) a 0
Cambia l'immagine dell'oggetto stesso (SELF, sempre la porta) a 0 (la
    porta chiusa)
Altrimenti
Mostra Il Testo "La porta è già chiusa"
```

La programmazione del nostro oggetto "door" è terminata. Premete il bottone OK per tornare al Doopsi Editor.

1.24 DOOPSI-GS - Coder Tutorial 2

Tutorial 2

In questo tutorial mostreremo come modificare le caratteristiche di un oggetto da un altro oggetto. Questo tutorial presuppone che voi abbiate già creato due oggetti, uno chiamato "lamp" con due immagini, una di lampadina spenta ("light_off") ed una accesa ("light_on"); e l'altro chiamato "button", con una sola immagine, di bottone.

- Selezionate l'oggetto "button" ed entrate nel Doopsi Coder.
- Premete sul bottone "Spingi", per editare il codice riferito alla pressione da parte del giocatore del bottone "button".
- Eseguite queste operazioni:
 - Comando "IfStatus"
 - Selezionate "lamp"
 - Inserite 0
 - Comando "ChangeStatus"
 - Selezionate "lamp"
 - Inserite 1
 - Comando "ChangeShape"
 - Selezionate "lamp"
 - Selezionate "light_on"
 - Comando "Else"
 - Comando "ChangeStatus"
 - Selezionate "lamp"
 - Inserite 0
 - Comando "ChangeShape"
 - Selezionate "lamp"
 - Selezionate "light_off"
 - Comando "EndIf"

Nella finestra di codice di Doopsi Coder, comparirà il seguente programma:

```
IfStatus lamp, 00
ChangeStatus lamp,01
ChangeShape lamp,01
Else
ChangeStatus lamp,00
ChangeShape lamp,00
EndIf
```

Che "tradotto" significa:

```
Se lo Status della lampada ("lamp") è 0 (la lampada è spenta)
Cambia lo Status di lamp in 1 (lampada accesa)
Cambia l'immagine di lamp a "light_on"
Altrimenti (la lampada è accesa)
Cambia lo Status di lamp a 0 (lampada spenta)
Cambia l'immagine di lamp a "light_off"
```

Come vedete dal codice, non abbiamo dovuto programmare l'oggetto "lamp", eppure l'oggetto modifica il suo stato (anche grafico) man mano che il giocatore preme il pulsante "button". Naturalmente, questo tutorial è molto semplice, ma è possibile creare dipendenze molto complesse tra

oggetti differenti, rendendo il gioco finale ricco e articolato.

1.25 DOOPSI-GS - Descrizione Generale del Coder

Descrizione generale.

Prima di tutto dovete selezionare l'azione che vi apprestate a programmare: per far ciò premete uno dei nove bottoni "Azioni Possibili" che compaiono nell'angolo in alto a sinistra della finestra; il nome dell'azione scelta comparirà nel campo "Azione Corrente" sul fondo della finestra.

Per inserire una qualsiasi istruzione eseguite le seguenti azioni: cliccate sul nome dell'istruzione desiderata contenuta nel lister che si trova alla destra delle "Azioni Possibili"; l'Editor vi chiederà di riempire qualsiasi argomento l'istruzione potrebbe avere. Dopo aver soddisfatto tutte le richieste potrete vedere apparire la linea di codice nell'area del programma (il grosso rettangolo con un linea evidenziata in esso). La barra nell'area del programma è una specie di cursore che evidenzia la linea corrente. L'istruzione che state editando verrà inserita prima di quella evidenziata. Potete muovere il cursore cliccando direttamente sull linea scelta oppure usando i gadget U (Up), D (Down), T (Top) e B (Bottom). Per eliminare una linea premete semplicemente il bottone "Uccidi".

I gadget:

Nella zona in alto a sinistra vi sono i nove gadget per selezionare una delle nove azioni da programmare. Alla destra di questa zona vi è il lister con tutte le istruzioni del linguaggio Doopsi. Gli altri gadget sono:

T	Sposta all'inizio del codice.
B	Sposta alla fine del codice.
U	Sposta in su di una linea. (Per muovervi su una linea potete cliccare direttamente su quella linea.)
D	Sposta in giù di una linea. (Per muovervi su una linea potete cliccare direttamente su quella linea.)
Uccidi	Rimuove la linea di codice attualmente evidenziata.

1.26 DOOPSI-GS - Anim Workshop

```
*****  
L' AnimWorkshop.  
*****
```

Questo editor vi permette di creare animazioni mettendo in sequenza alcune

immagini. Il risultato è che potete mutare un semplice Oggetto in qualcosa di più complesso chiamato AnimObject, ovvero un oggetto contenente un'animazione. Tale animazione verrà visualizzata quando l'AnimObject è disegnato sulla scena.

Tutorial 1
Tutorial 2

Descrizione Generale di Anim Workshop

1.27 DOOPSI-GS - Anim Workshop Tutorial 1

- Create un nuovo oggetto con "Pendolo" come nome interno.
- Attivate AnimWorkshop.
- Premete il gadget "Carica Frames" e, utilizzando il file requester, selezionate l'immagine denominata "pendulum.000". AnimWorkshop caricherà le tre immagini chiamate "pendulum.000", "pendulum.001" e "pendulum.002".
- Selezionate il fotogramma numero 0 e cliccate sul gadget "Agg Frame"; selezionate il fotogramma numero 1 e cliccate sul gadget "Agg Frame"; ripetete il tutto anche per il terzo ed ultimo fotogramma.
- Premete il gadget "Ping-Pong" per creare un'animazione di tipo Ping-Pong (nel nostro caso verrà aggiunto un solo fotogramma).
- Scrivete 3 nel gadget del "Frame Rate".
- Premete "Anima" per vedere l'animazione e il tasto destro del mouse per fermarla.
- Premete il gadget "Ok" per uscire da AnimWorkshop.

1.28 DOOPSI-GS - Anim Workshop Tutorial 2

Ed eccovi un tutorial più difficile su "come costruire l'Omino":

- Per prima cosa create un nuovo oggetto utilizzando l'editor degli Attributi (suppongo che sappiate già fare questo) e chiamatelo (nome interno) "OminoDestra". Ora siete pronti per attivare AnimWorkshop.
 - Premete il bottone "Carica Frames" e, utilizzando il file requester, selezionate l'immagine chiamata "man_walk_right.000": AnimWorkshop caricherà questa immagine e tutte le seguenti col medesimo nome e numero crescente; questi fotogrammi saranno disegnati su un nuovo schermo. Potete scorrere le immagini utilizzando i gadget "Prec" e "Succ" nella parte alta della finestra.
 - Posizionatevi sul fotogramma numero 1 (il fotogramma 0 è quello "a riposo" e lo prenderò in considerazione più tardi) e premete il bottone "Agg Frame": il numero 1 apparirà nella finestra della Sequenza. Ora
-

muovetevi sul fotogramma numero 2 e premete ancora su Agg Frame; ripetete queste azioni fino a quando anche l'ultimo fotogramma farà parte della sequenza.

- Ora potete inserire il frame rate nel gadget "Frame Rate": scrivete 12 e premete Return.

- La vostra animazione è completa: premete il gadget "Anima" per ammirarla.

- Fermate l'animazione premendo il tasto destro del mouse. Ora potete uscire da AnimWorkshop premendo il bottone "Va Bene" o "Cancella" a seconda se siete soddisfatti della vostra animazione o meno.

- E il frame numero 0? Non si può includerlo nell'animazione perché è il fotogramma "a riposo", ma il Player ha bisogno di sapere quale frame usare quando l'Omino è fermo: quindi noi lo agganciamo al nostro oggetto Omino e puntiamo ad esso mediante il campo "Immagine Corrente" nell'editor degli Attributi. (Ovviamente questo campo non ha significato alcuno per una animazione qualunque.)

- Un'ultima nota: se ora attivate l'editor degli Attributi vedrete tutte le immagini agganciate al nostro AnimObject mediante AnimWorkshop. Questo significa che potete agganciare i fotogrammi anche utilizzando l'editor degli Attributi se preferite.

1.29 DOOPSI-GS - Descrizione Generale di Anim Workshop

Descrizione generale.

A questo punto suppongo che sappiate già come creare un nuovo oggetto. Se volete convertire tale oggetto in un AnimObject allora attivate l'AnimWorkshop. Se l'utente ha già agganciato alcune immagini a questo oggetto utilizzando l'editor degli Attributi allora tali immagini saranno caricate e visualizzate immediatamente; se una o più immagini non vengono trovate l'editor segnalerà un errore e le immagini mancanti rimarranno vuote. In alternativa è possibile caricare le immagini mediante AnimWorkshop stesso: ma si tenga presente che in questo caso le immagini devono costituire una sequenza sul disco, vale a dire che devono avere tutte il medesimo nome seguito da un punto e da un numero a tre (tre, non quattro o due!) cifre, ad es. "pippo.000". Quindi, quando l'utente seleziona il gadget Carica Frames, AnimWorkshop aprirà un file requester: l'utente deve selezionare la prima immagine della sequenza (che non deve necessariamente essere quella numerata "pippo.000"; potete iniziare dall'immagine "pippo.154" se vi piace), ed allora l'editor proseguirà caricando tutte le immagini che trova con il medesimo nome e con numero crescente (ad es.: "pippo.154", "pippo.155", e così via) fino a quando raggiunge l'ultima o non ha più spazio per agganciare ulteriori immagini all'oggetto corrente (il numero di immagini che un oggetto può possedere è ovviamente limitato).

Dopo che avete caricato le immagini siete pronti per costruire la sequenza dei fotogrammi. Prima di tutto, vi sono due gadget nella parte alta della finestra, chiamati "Prec" e "Succ", per scorrere i fotogrammi, e fra di essi è scritto il numero sequenziale del fotogramma visualizzato: potete

inserirlo manualmente. Dunque dovete agire come segue: selezionate il fotogramma da includere nella sequenza e premete il bottone Aggiugi Frame. Semplice, no? Il fotogramma verrà inserito subito dopo la posizione corrente, e voi saprete sempre qual è la posizione corrente in quanto è evidenziata: potete inoltre muovere questa specie di cursore utilizzando i gadget "Prec" e "Succ" alla base della finestra o utilizzando i tasti cursore. A proposito, se occorre inserire un fotogramma prima del primo (scusate il gioco di parole, ma rende l'idea) dovete muovere il cursore sul primo fotogramma della sequenza e poi muoverlo ancora una volta verso sinistra: il cursore sparirà fuori dalla finestra della Sequenza e siete pronti per cliccare sul gadget "Aggiungi Frame". Posso dirvi di più: se cliccate due volte sul numero di un fotogramma nella finestra della Sequenza tale fotogramma sarà visualizzato. Rimuovere un fotogramma è ancora più semplice: posizionatevi sul fotogramma incriminato nella Sequenza e cliccate sul bottone "Rimuovi Frame".

Veniamo ora alla descrizione dei gadget rimanenti. Un modo comodo per creare un'animazione a ping-pong è quello di costruire la parte "ping" come descritto precedentemente e poi selezionare il gadget "Crea Ping-Pong": AnimWorkshop aggiungerà alla vostra sequenza la parte "ping" ribaltata (senza il primo e l'ultimo fotogramma) per creare la parte "pong". Inoltre, se non siete soddisfatti della vostra sequenza potete sempre eliminarla con una sola pressione del bottone "Cancella Sequenza". Potete comunicare ad AnimWorkshop la velocità della vostra animazione modificando il valore del gadget "Frame Rate". Ultimo ma non meno importante è il gadget "Anima". So che sapete a cosa serve questo, ma lasciatemi aggiungere un'altra cosa: Doopsi visualizza tutte le animazioni in loop, ovvero la sequenza viene letta da sinistra a destra fino alla fine, e poi ancora dall'inizio fino a quando non viene fermato.

I gadget:

Carica Frames	Carica una sequenza di fotogrammi. Una sequenza è composta da files con lo stesso nome ma con numero sequenziale crescente (ad es.: pippo.000, pippo.001, ...). Voi selezionate il primo nome della lista.
Prec	Muove al fotogramma precedente nella lista.
Succ	Muove al fotogramma successivo nella lista.
Agg Frame	Aggiunge alla sequenza il fotogramma attualmente visualizzato.
Canc Frame	Cancella il fotogramma selezionato dalla sequenza. Voi selezionate il fotogramma muovendo il cursore sopra di esso.
Crea Ping-Pong	Crea un'animazione a ping-pong modificando la vostra sequenza di fotogrammi.
Frame Rate	Cambia il frame rate dell'animazione.
Cancella Sequenza	Elimina la sequenza di fotogrammi.
Anima	Mostra l'animazione.

Prec Va al fotogramma precedente nella sequenza (potete usare anche i tasti cursore).

Succ Va al fotogramma successivo nella sequenza.

1.30 DOOPSI-GS - Dialog Editor

```
*****  
L'Editor dei Dialoghi.  
*****
```

Introduzione.

In ogni avventura che si rispetti, devono essere presenti dei dialoghi tra più personaggi. Migliori sono i dialoghi, più brillanti sono le conversazioni, più divertente risulta l'avventura. Creare i dialoghi non è un compito facile da eseguire: bisogna tenere in considerazione moltissimi elementi e prevedere una risposta differente ad ogni frase che è possibile dire. E' per questo motivo, che vi consigliamo vivamente di pianificare attentamente i vostri dialoghi e di tracciare su carta dei veri e propri diagrammi nei quali descrivere in maniera molto "visiva" come dovrebbe essere strutturato un dialogo.

In Doopsi i dialoghi possono essere strutturati in maniera molto potente e versatile ed è possibile scrivere dialoghi che richiamano altri dialoghi o che saltano da una parte all'altra per poi tornare sui loro stessi passi.

Il Dialog Editor è uno strumento semplice e funzionale che vi permetterà di creare dialoghi complessissimi in un brevissimo spazio di tempo: attenzione però, perché, come vi abbiamo già detto, se non pianificherete prima tutto il dialogo, rischiate veramente di perdere, per così dire, il filo del discorso.

Cos'è un dialogo:

Un Dialogo, in DOOPSI, è formato da una sequenza più o meno lunga di pagine. Come per tutti gli altri elementi di Doopsi anche le pagine di dialogo hanno il loro proprio "nome interno" che le identifica univocamente.

Per il resto, una pagina è formata da una "risposta", che è la frase che dirà l'interlocutore all'Omino, e una o più frasi selezionabili dall'utente, come risposta alla "risposta"... hmmm, detto così può sembrare un pò confuso, cerchiamo di spiegare meglio, con un esempio: iniziamo subito con il tutorial.

Tutorial 1

Dialog Editor General Description

1.31 DOOPSI-GS - Dialog Editor Tutorial 1

Tutorial 1 - Un Breve Dialogo

Per entrare nel Dialog Editor, premete semplicemente il bottone "Dialog Editor" presente nella Finestra Principale.

- Premete il bottone "Nuova" per creare una nuova pagina di dialogo. Vi verrà mostrato un requester per l'inserimento del nome della pagina. Dal momento che questa pagina verrà utilizzata come punto di partenza, sarà meglio darle un nome significativo, scrivete quindi "Partenza_Dialogo". Adesso siete in grado di iniziare a scrivere le frasi vere e proprie.
- Premete il bottone "Agg." in basso a sinistra, per aggiungere una nuova frase. La stringa "Nuova Frase" comparirà nel gadget chiamato, per l'appunto, "Frase:". Cancellate "Nuova Frase" ed inserite invece questo: "Ciao, come stai?"
- Ripetete adesso il punto 2, inserendo però i messaggi: "Come va!" e "Adesso devo proprio andare".
- Adesso dovremo creare una seconda pagina, premiamo "Nuova" e mettiamo come nome "Come_Stai", così ci ricorderemo che quello che immetteremo in questa pagina sarà in risposta a "Ciao, come stai?" della pagina precedente.
- Nel gadget in alto, dove c'è scritto "Risposta:", scrivete "Bene!", questa è la risposta che il nostro interlocutore ci darà. Scrivere qualcosa in questo campo, non è strettamente necessario, ma è sempre meglio fare in modo che il nostro interlocutore dica qualcosa... altrimenti il nostro dialogo si potrebbe trasformare in un "monologo".
L'unica situazione nella quale si può effettivamente omettere una stringa di risposta, è nella prima pagina del nostro dialogo, quando, cioè, "attachiamo bottone" con chi vogliamo parlare.
- Ripetiamo il passo 2 e scriviamo solo una frase che dica "Ora vado, ciao".
- Adesso dobbiamo preoccuparci di "agganciare" le due pagine di dialogo. Per farlo, torniamo alla prima pagina di dialogo. Premete il gadget "Seleziona": comparirà un Lister con i nomi delle pagine di dialogo, nel nostro caso, ci saranno due nomi: "Partenza_Dialogo" e "Come_Stai". Selezionate "Partenza_Dialogo".
La finestra del Dialog Editor, mostrerà di nuovo la pagina che abbiamo editato in precedenza.
- Adesso premete sulla frase "Ciao, Come Stai?", che comparirà nel gadget di stringa (se volete, potrete anche editarla e modificarla) e poi premete sul bottone chiamato "Salto:", apparirà lo stesso Lister che abbiamo appena incontrato. Questa volta, selezionate la frase "Come_Stai": il nome di questa pagina comparirà nel gadget di stringa "Salto:"
Ecco fatto! Adesso, quando, durnate il gioco, l'utente selezionerà la frase "Ciao! Come Stai?", l'interlocutore risponderà "Bene!".

E' stato facile, no?

- Giusto per sport, agganciamo anche alla frase "Come Va?", la pagina "Come_Stai". Premete su "Come Va?", poi su "Salto:" e selezionate "Come_Stai".

NOTA: una stringa Senza salto significa FINE DEL DIALOGO. Quando l'utente seleziona una di queste frasi, il dialogo termina.
Nel nostro esempio abbiamo ben due frasi di questo tipo: una è "Adesso devo proprio andare" nella prima pagina e "Ora vado, ciao" nella seconda pagina.

NOTA2: per convenzione, la frase che fa "uscire" dalla modalità di dialogo è l'ultima tra le scelte possibili. Dove potete, mantenete inalterata questa convenzione.

1.32 DOOPSI-GS - Descrizione Generale del Dialog Editor

Descrizione generale.

Il Dialog Editor è utilizzato per creare dialoghi fra l'Omino ed un'altro Oggetto sulla scena. Come abbiamo avuto occasione di dire introducendo questo editor, un dialogo è costituito da un numero di pagine agganciate fra di loro; qualunque pagina può essere la prima pagina: il nome della prima pagina verrà passato come argomento all'istruzione ShowDialog nel Doopsi Coder. Tutte le altre pagine del dialogo seguono automaticamente poiché sono agganciate assieme mediante il campo "Salto".

Descriviamo ora il contenuto di una pagina. In ciascuna pagina vi è un campo "Risposta" che contiene la frase che l'Oggetto dice quando è il suo turno di parlare: questo è il primo campo che il Player guarda quando visualizza un dialogo. Questo implica che se la prima pagina ha il campo "Risposta" non vuoto, allora la conversazione viene iniziata dall'Oggetto. Quando l'Oggetto ha finito di parlare (ovvero, quando l'utente, durante il gioco, ha premuto il bottone del mouse per cancellare il testo che appare sulla scena), tutte le frasi contenute nel grosso lister sotto il campo "Risposta" vengono visualizzate nello spazio riservato alla console. Il Player attende che l'utente selezioni una frase e la visualizza sulla scena (aspettando ancora che l'utente cancelli il testo prima di proseguire). Poi il Player esamina il contenuto del campo "Salto": se non è vuoto, il Player cerca una pagina con il nome uguale alla stringa contenuta nel campo "Salto". Alla fine, se questa pagina è stata trovata il processo ricomincia da capo: viene visualizzata prima la risposta, poi le scelte al di sotto di essa, e così via. Un dialogo incontra la propria fine quando il campo "Salto" della frase selezionata è vuoto oppure la pagina agganciata non viene trovata.

Per costruire un dialogo dovete prima creare alcune pagine. Il bottone "Nuova" sulla destra crea una nuova pagina e chiede il nome da attribuire a tale pagina: questo nome è l'identificatore che Doopsi usa per indirizzare la pagina, quindi dovete fare attenzione a non utilizzare due volte lo stesso nome. Per modificare la risposta, scrivetela semplicemente nel gadget stringa "Risposta". Per aggiungere una nuova frase dovete premere il bottone "Agg.": il testo "Nuova Frase" apparirà sia nel lister, sia nel gadget stringa sotto di esso. Ora potete modificare tale frase scrivendola

nel gadget stringa "Frase": il nuovo testo rimpiazzerà quello vecchio nel lister. Nel campo "Salto" va inserito il nome della pagina a cui saltare quando la frase corrente viene selezionata durante il gioco: potete modificare tale nome inserendolo a mano nel gadget stringa oppure premendo il bottone "Salto"; quest'ultimo apre un lister con tutti i nomi delle pagine create fino ad ora: cliccate sul nome desiderato e poi sul gadget "Va Bene".

Ora descriviamo i rimanenti gadget nella fila in basso. Il bottone "Memorizza" immagazzina il contenuto del campo "Frase" nel lister (è lo stesso che premere il tasto Return nel campo "Frase"). Il gadget "Seleziona" vi permette di muovervi attraverso le pagine: quando lo premete appare un lister con i nomi delle pagine create fino ad ora, e potete selezionare quella da visualizzare. Il gadget "Rimetti" annulla l'ultima modifica apportata al campo "Frase". Il bottone "Canc" rimuove la frase correntemente selezionata.

I gadget nella colonna a destra hanno i seguenti significati. Il gadget "Nuova" crea una nuova pagina. Il bottone "Salva" salva tutte le pagine create sino ad ora, e il gadget "Carica" carica un insieme di pagine salvate con il bottone precedente. Il gadget "Nome Pag." vi permette di cambiare il nome della pagina corrente. Il bottone "Uccidi Pag." rimuove solo la pagina corrente e il bottone "Uccidi Tutto" elimina tutte le pagine create.

I Gadget:

Iniziamo con i gadget a destra sullo schermo:

Nuova	Crea una nuova pagina.
Carica	Carica un insieme di pagine.
Salva	Salva le pagine create.
Nome Pag.	Modifica il nome della pagina attuale.
Uccidi Pag.	Elimina la pagina corrente.
Uccidi Tutto	Elimina tutte le pagine create.

Ed ecco i bottoni in basso:

Agg.	Aggiunge una nuova frase.
Memorizza	Memorizza la pagina corrente.
Seleziona	Permette di selezionare una pagina di dialogo.
Rimetti	Rimette l'ultima frase modificata.
Canc	Cancella la frase attuale.

1.33 DOOPSI-GS - Preferenze

```
*****  
Preferenze.  
*****
```

Doopsi è un programma complesso e flessibile ed alcune delle sue caratteristiche possono essere adattate alle esigenze del programmatore. In Doopsi, le modifiche possono essere effettuate tramite la finestra di Preferences, suddivisa in quattro sezioni, che esamineremo singolarmente.

Selezione dei Percorsi.

In questa sezione è possibile scegliere le directory contenenti tutti i file di un determinato tipo. Creando delle avventure complesse, è sempre meglio suddividere grafica, suoni e tutto il resto in varie directory. Con questi percorsi, ogni volta che sarà necessario un file requester di un determinato tipo (ad esempio per caricare un fondale) la directory verrà automaticamente impostata a quella inserita nelle preferenze. Nel caso non si desideri utilizzare alcune delle directory di default, sarà sufficiente inserire all'interno del campo una stringa vuota.

Il secondo gruppo di percorsi di default è dedicato al Player: essi servono al Player per trovare i file che sono indispensabili per iniziare il gioco, vale a dire il file contenente la console (default: "Doopsi:Data/panel.iff"), il file "player.dat" (default: "Doopsi:Data/player.dat") ed infine (opzionale) il catalogo per un linguaggio diverso dall'inglese (default: "Doopsi:Data/player.catalog"). In questi string gadget occorre quindi inserire, oltre al percorso completo, anche il nome dei file. Questo permette, ad esempio, di utilizzare una console diversa per ciascuna avventura.

L'ultimo string gadget, chiamato "Player", contiene il percorso completo ed il nome del programma "DoopsiPlayer": questo serve all'Editor per trovare il Player quando viene premuto il bottone "Prova Gioco" nella Finestra Principale.

A fianco del gadget di stringa contenente il path, c'è un piccolo gadget con scritto "R" (ricerca), che aprirà un path requester con il quale sarà possibile cercare la directory che desideriamo impostare come default.

Manipolazione dei Colori.

In questa sezione è possibile caricare, editare e salvare i colori dell'interfaccia dell'editor di Doopsi. Premendo "Carica" sarà possibile caricare i primi quattro colori di una qualsiasi immagine IFF, con "Salva" potremo salvare la palette e con "Edita" apparirà un Palette requester con il quale sarà possibile modificare direttamente i colori.

Scelta dei nomi di Default.

Questa è forse la sezione più importante delle preferenze. Come dovreste sapere, Doopsi utilizza per il personaggio dell'avventura dei nomi di default necessari perché il Player utilizzi le animazioni corrette quando necessario. Potete trovare la spiegazione di come questo meccanismo funzioni nella sezione "Alcune note su come il Player trova l'Omino" posta

immediatamente dopo la sezione del manuale relativa alla Finestra Principale. Seguono i nomi dei campi ed il loro significato:

NOME	VALORE DEF.	DESCRIZIONE
Root	Man	E' il nome "base" mediante il quale sono definiti i nomi delle animazioni dell'Omino.
Left	L	Questi quattro valori determinano le animazioni del personaggio nelle varie direzioni. Ad es. "ManU" significa "animazione Omino che cammina verso l'alto"
Right	R	
Up	U	
Down	D	
Talk	Talk	Questa è una specie di "seconda root" ed è il suffisso che viene aggiunto DIRETTAMENTE alla Root vera e propria per definire le animazioni del personaggio mentre parla. A questa root verrà poi aggiunto normalmente un suffisso Left, Right, Up o Down. Per es. il personaggio che parla a sinistra è definito così: ManTalkL.

Questi nomi, essendo solo "interni" al Player, potevano anche rimanere "rigidi" e non essere modificabili, ma abbiamo preferito permetterne la modifica, per poter rendere tutto ancora più leggibile. Infatti, potrete modificare questi nomi, assegnando valori più significativi. Facciamo un esempio:

```
Root = Uomo
Talk = _che_parla
L     = _a_sinistra
R     = _a_destra
```

Il famigerato ManTalkL diventerà "Uomo_che_parla_a_sinistra", sicuramente più elegante. Ma attenzione, questo è solo un esempio: i nomi interni degli Oggetti in Doopsi possono essere di massimo 30 caratteri.

Scena di Partenza.

Con questa ultima parte della finestra di Preferences è possibile definire il nome della prima scena che verrà mostrata all'avvio dell'avventura.

I bottoni USA/CARICA/SALVA/ANNULLA

I bottoni presenti in fondo alla finestra di Preferences permettono di usare le preferenze appena inserite ("USA"), salvarne le preferenze attuali ("SALVA"), caricare una configurazione precedentemente salvata ("CARICA") e di annullare i nuovi settaggi ("ANNULLA").

Nota: le preferenze vengono SEMPRE salvate in un file completo di Doopsi, per permettere la gestione di più avventure differenti contemporaneamente con settaggi diversi.

ToolTypes:

Dalla versione 1.10, Doopsi supporta alcuni tooltypes. Al momento riconosce i seguenti:

SCREENMODE=\$xxxxx	Per selezionare il tipo di schermo. (Ad es.: \$8000 = Alta ris. non interlacciato)
PREFS=ON OFF	Se avete problemi con i colori dell'Editor, con questo potete evitare che l'Editor di preferenze modifichi i colori dell'interfaccia.

1.34 DOOPSI-GS - Appendice A

```
*****
Appendice A: Istruzioni Doopsi.
*****
```

Nella seguente lista oggetti e scene sono identificati utilizzando il loro nome interno, quindi Oggetto rappresenta il nome interno dell'Oggetto stesso, e così via.

AddStatus Oggetto, Valore

Aggiunge Valore al campo di stato dell'oggetto selezionato. Il valore totale dello stato di un oggetto è compreso fra -32767 e +32768.

ChangeDescription Oggetto, Testo

Cambia il nome dell'oggetto in Testo. Questo nome è quello che appare sulla linea di testo della console quando l'oggetto viene selezionato sulla scena.

ChangeScene Scena

Se Scena è stata trovata questa istruzione esce dalla scena corrente e visualizza la nuova scena.

ChangeShape Oggetto, Immagine

Cambia l'immagine attuale dell'Oggetto selezionato. Se l'Oggetto è visualizzato sulla scena la nuova immagine verrà disegnata e rimpiazzerà quella vecchia. Se l'oggetto è nell'inventario questo comando modificherà l'immagine di inventario (e la ridisegnerà). Infine, se l'Oggetto non è visualizzato questa istruzione modificherà solo il valore dell'immagine attuale dell'Oggetto stesso. Questa istruzione non funziona con gli AnimObject o con le Zone Vuote (per ovvi motivi) quando sono disegnate sulla scena, ma le loro immagini di inventario potranno comunque essere modificate. Con ChangeShape l'immagine attuale dell'Oggetto è cancellata perché la nuova immagine viene ridisegnata immediatamente sopra essa: questo implica che le due immagini devono avere le medesime dimensioni.

ChangeShapeBG Oggetto, Immagine

Questo comando è simile a ChangeShape, ma prima di disegnare la nuova immagine il Player cancella l'immagine attuale dell'Oggetto ridisegnando lo sfondo. Questo elimina la limitazione presente in ChangeShape che la nuova immagine debba avere dimensioni uguali o maggiori di quella vecchia.

ChangeStatus Oggetto, Valore

Scrivi Valore nel campo di stato dell'Oggetto selezionato. Lo stato può essere 0 o 1 per rappresentare una porta chiusa o aperta rispettivamente, e così via... Se l'oggetto è una moneta questo campo può essere usato per memorizzare il numero di monete che l'Omino possiede. Il valore che il campo di stato di un oggetto può contenere va da -32767 a +32768.

Close Oggetto

Si veda l'istruzione Open.

ConsoleHide

Spegne la console.

ConsoleShow

Accende la console.

Drop Oggetto

Si veda l'istruzione Open.

Else

Else viene usato in un gruppo If: se la condizione è falsa allora il Player eseguirà tutte le istruzioni che seguono il comando Else fino al più vicino EndIf (o fino alla fine del programma se quest'ultimo non è presente).

EndIf

Questa istruzione termina un gruppo If. Potete avere fino a 10 gruppi If nidificati.

EraseInventory Oggetto

Rimuove l'Oggetto dalla lista dell'inventario e cancella la sua immagine di inventario dalla console.

EraseObject Oggetto

Rimuove l'Oggetto dalla scena. Questa istruzione non si applica agli AnimObject: in questo caso l'utente deve utilizzare StopAnim.

GetInventory Oggetto

Aggiunge l'Oggetto alla lista dell'inventario e disegna la sua immagine di inventario attuale sulla console. Questa istruzione fallisce se la lista dell'inventario è piena, in quanto essa può contenere solo un numero limitato di oggetti.

IfOnScene Oggetto

Restituisce Vero se l'Oggetto è disegnato sulla scena.

IfSceneIs Scena

Restituisce Vero se l'Omino è attualmente nella scena Scena.

IfStatus Oggetto, Stato

Restituisce Vero se il campo di stato dell'Oggetto selezionato è uguale a Stato.

IfUsedWith Oggetto

Restituisce Vero se l'oggetto corrente è usato con Oggetto. Questo significa che avete prima selezionato il bottone Usa e cliccato una volta su Oggetto e poi, dopo che il testo "Usa Oggetto con..." è apparso sulla console, avete cliccato due volte sull'oggetto corrente: questo attiva il programma Doopsi relativo all'azione Usa dell'oggetto corrente dove dovrebbe trovarsi questa istruzione. Affinché cliccando su Oggetto appaia la scritta "con..." ed il Player attenda la selezione di un altro oggetto è necessario che Oggetto abbia UseWith come prima istruzione del programma relativo all'azione Usa.

InvToScene Oggetto

Preleva l'Oggetto selezionato dall'inventario e lo disegna sulla scena alle coordinate contenute in Oggetto: tali coordinate sono inizializzate quando l'Oggetto viene posizionato sulla scena mediante l'Editor. Se l'oggetto è un AnimObject l'animazione verrà automaticamente visualizzata.

InvToSpot Oggetto, Spot

Uguale a InvToScene, ma l'Oggetto verrà disegnato alle coordinate dello Spot.

Look Oggetto

Si veda l'istruzione Open.

ManDirection Estensione

Forza l'Omino a girarsi nella direzione selezionata: Estensione è una delle quattro estensioni indicanti una direzione che l'utente può modificare mediante l'editor di preferenze.

ManTalkColour Valore

Con questa istruzione l'utente sceglie il registro colore da usare per il testo in tutte le successive istruzioni ShowText o ShowDialog quando l'Omino parla.

MoveMan Spot

Questa istruzione permette all'utente di forzare l'Omino a muoversi verso lo Spot desiderato sulla scena.

MoveObject Oggetto, Spot

Muove l'Oggetto dalla sua posizione attuale sulla scena alla posizione specificata da Spot. Durante il movimento un AnimObject verrà sostituito all'immagine corrente dell'Oggetto se tale AnimObject è stato precedentemente selezionato mediante l'istruzione SetMoveAnim.

MoveObjectOnPath Oggetto, Spot

Questo è simile a MoveObject con la differenza che l'Oggetto selezionato si muoverà seguendo il percorso della scena corrente e si fermerà al nodo più vicino allo Spot selezionato. Un'altra leggera differenza è che si suppone che l'hot-spot dell'oggetto (o, se presente, dell'animazione che lo sostituisce) sia posizionato nell'angolo in basso a sinistra dell'immagine.

MoveObjectToObject Oggetto1, Oggetto2

Questo è simile a MoveObject, e muove Oggetto1 alla posizione specificata da Oggetto2.

MoveObjToObjOnPath Oggetto1, Oggetto2

Questo è simile a MoveObjectOnPath, e muove Oggetto1 alla posizione specificata da Oggetto2.

MusicPause

Sospende momentaneamente la musica. Per farla ripartire basta utilizzare un'altra volta questo comando.

MusicPlay File

Suona il contenuto del File. File deve essere un modulo tracker standard.

MusicStop

Interrompe la musica.

ObjTalkColour Valore

Questo è simile a ManTalkColour, ma vi permette di scegliere il registro colore del testo dell'oggetto a cui l'Omino sta parlando durante una istruzione ShowDialog.

Open Oggetto
Close Oggetto
Look Oggetto
Take Oggetto
Drop Oggetto
Push Oggetto
Pull Oggetto
Talk Oggetto
Use Oggetto

Queste istruzioni eseguono il codice Doopsi contenuto nella corrispondente azione di Oggetto. Esse sono gestite come sottoprogrammi, e potete nidificare fino ad un massimo di 10 sottoprogrammi.

Pull Oggetto

Si veda l'istruzione Open.

Push Oggetto

Si veda l'istruzione Open.

PutObject Oggetto, Spot

Disegna l'Oggetto alla posizione dello Spot. Oggetto può anche essere una Zona Vuota.

SceneToInv Oggetto

Preleva l'Oggetto selezionato dalla scena e lo ripone nell'inventario. Se

l'oggetto è un AnimObject l'animazione verrà automaticamente fermata.

SetManName Oggetto

Questa istruzione serve per cambiare le animazioni dell'Omino: la radice del nome (cioè dopo aver rimosso qualunque estensione) dell'Oggetto è utilizzata per costruire i nomi dell'Omino fino a quando SetManName viene chiamata nuovamente o è invocata da un nodo del percorso.

SetMoveAnim AnimObject

L'AnimObject selezionato verrà usato come animazione per la seguente istruzione MoveObject o simili (MoveObjectOnPath, MoveObjectToObject, MoveObjToObjOnPath). Se il nome dell'AnimObject termina con una delle quattro estensioni (indicanti le quattro direzioni) allora il Player è autorizzato a scegliere la direzione opportuna prima di muovere l'oggetto: se tale direzione non è fornita verrà utilizzata l'immagine corrente dell'oggetto, e quindi la selezione dell'animazione sarà totalmente ignorata.

SetTalkAnim AnimObject

Questa istruzione comunica al Player che, quando il prossimo ShowDialog verrà eseguito, deve usare AnimObject per mostrare che l'oggetto selezionato sta parlando. Se l'utente termina il nome interno di AnimObject mediante una delle quattro estensioni indicanti una direzione (tali estensioni possono essere modificate usando l'editor di preferenze), allora il Player rimuove questa estensione da tale nome interno e sceglie la direzione appropriata (dipendente dalla posizione relativa dell'oggetto rispetto all'Omino). Poi aggancia l'estensione relativa alla direzione scelta al nome interno di AnimObject e cerca l'oggetto risultante. Se questo viene trovato l'animazione corrispondente sarà visualizzata quando l'oggetto parla.

ShowAnim AnimObject

Questa istruzione visualizza l'animazione contenuta in AnimObject. Tale animazione verrà posizionata alle coordinate interne di AnimObject: queste coordinate vengono inizializzate quando l'AnimObject è posizionato sulla scena mediante l'Editor (e rimangono memorizzate anche se l'AnimObject viene rimosso dalla scena in un secondo tempo). Il numero di animazioni che il Player può visualizzare contemporaneamente è limitato.

ShowAnimAtSpot AnimObject, Spot

Visualizza l'animazione contenuta in AnimObject alle coordinate di Spot.

ShowDialog Dialogo

Questo comando inizia una conversazione tra l'Omino e l'oggetto selezionato. Dialogo è la prima pagina della conversazione. Se

l'animazione che mostra l'Omino che parla viene trovata essa sarà visualizzata; potete scegliere un'animazione che mostra l'oggetto che parla mediante SetTalkAnim. I colori dei testi dell'Omino e dell'oggetto vengono scelti mediante le istruzioni ManTalkColour e ObjTalkColour rispettivamente.

ShowPicture File

Nasconde momentaneamente la scena e la console per visualizzare il contenuto del File. File deve essere un'immagine Iff ILBM. Se l'immagine è più grande dell'area dello schermo potete muoverla spostando il mouse contro i bordi dello schermo. Per ritornare allo stato precedente premere il tasto sinistro del mouse.

ShowText Testo

Visualizza Testo. La posizione e la suddivisione in linee del testo verranno determinate automaticamente mediante la posizione dell'Omino sulla scena e la larghezza della scena stessa. La direzione dell'Omino determina quale delle quattro animazioni verrà visualizzata, se presente, per far parlare l'Omino; se l'animazione non è presente il Player non si lamenterà e si limiterà a lasciare immutato l'Omino. Il colore del testo può essere cambiato utilizzando l'istruzione ManTalkColour. Per cancellare il testo è necessario premere il tasto sinistro del mouse.

SoundPlay File

Suona il contenuto del File. File deve essere un campionamento Iff 8SVX standard.

StopAnim AnimObject

Ferma l'animazione contenuta in AnimObject e la rimuove dalla scena.

Take Oggetto

Si veda l'istruzione Open.

Talk Oggetto

Si veda l'istruzione Open.

Use Oggetto

Si veda l'istruzione Open.

UseWith

Questa istruzione deve essere la prima nel programma dell'azione Usa di un

oggetto: essa comunica al Player che l'oggetto corrente deve essere usato insieme ad un altro oggetto; dunque il Player visualizza il testo "Usa Oggetto con..." e aspetta che l'utente selezioni un altro oggetto. Si veda anche IfUsedWith.

1.35 DOOPSI-GS - La Console

```
*****  
Appendice B: La console.  
*****
```

Lo schema della console incluso nel pacchetto di Doopsi mostra le diverse zone controllate dal Player: ci sono i nove bottoni delle azioni sulla sinistra, due frecce verticali per far scorrere l'inventario e, fra queste, un piccolo bottone usato per accedere all'interfaccia per il disco; la parte destra è occupata dall'inventario e, ultimo ma non meno importante, il rettangolo all'estremità superiore è la linea di testo dove appariranno tutti i messaggi. A proposito: ciascun quadratino dell'inventario misura $32 * 32$ pixel, quindi l'utente deve tener conto di queste dimensioni quando progetta le immagini di inventario per i suoi oggetti. Sotto la console, sullo stesso schermo è rappresentata l'interfaccia per il disco: l'area più vasta sulla sinistra è quella in cui comparirà la lista dei file e sotto di essa vi è lo spazio dove l'utente inserisce il percorso di ricerca dei file stessi; poi vi sono due frecce verticali per scorrere la lista dei file e cinque bottoni.

Lo schermo contenente la console e l'interfaccia disco deve essere in bassa risoluzione non interlacciato e l'utente può disegnarci sopra tutto quello che desidera, ma deve ricordare di rispettare le dimensioni delle zone delimitate. Si ricordi anche che i quadratini dell'inventario, la linea di testo e l'area dedicata alla lista dei file vengono riempite dal Player con il colore numero 0. Un altro suggerimento: tutte le immagini di inventario devono avere la medesima palette, quella della console, in quanto appariranno sulla console e quest'ultima rimane fissa per tutta la durata del gioco; invece le altre immagini possono avere palette differenti poiché è permesso cambiare i colori passando da una scena all'altra.

L'interfaccia del disco.

E' come tutti gli altri requester a cui siete abituati: il percorso di ricerca viene inserito nel gadget stringa situato nella parte bassa dello schermo e i nomi dei file vengono elencati nell'area sopra di esso. Per caricare un file l'utente deve selezionarne il nome con il tasto sinistro del mouse e poi cliccare sul bottone Carica alla destra dello schermo. Per salvare un file l'utente deve selezionarne il nome e poi cliccare sul bottone Salva: apparirà un cursore per permettere all'utente di modificare il nome del file; quando le eventuali modifiche sono state effettuate basta premere Return per salvare. Per salvare una nuova partita cliccate su Salva come prima, senza selezionare alcun nome.

I nomi che compaiono nel lister ed il percorso di ricerca sono limitati ad una lunghezza di 25 caratteri (penso sia abbastanza per la maggior parte delle applicazioni).

I cinque bottoni sulla destra sono denominati, in ordine: Carica, Salva, Gioca, Workbench ed Esci. I bottoni Carica e Salva sono già stati descritti. Il bottone Gioca permette di ritornare allo schermo della console e di continuare il gioco. Il bottone denominato Workbench attiva il multitasking: il Player scompare e l'utente può tornare alla partita interrotta selezionando la voce Doopsi-GS nel menu Tools del Workbench. Il bottone Esci..., lascio a voi immaginare lo scopo di quello.

1.36 DOOPSI-GS - Dentro DOOPSI

```
*****
Appendice C:  Dentro Doopsi.
*****
```

Nel seguito userò il termine Doopsi quando descriverò come funzionano alcune parti comuni all'Editor ed al Player.

Struttura degli oggetti di Doopsi.

Per capire come Doopsi agisce e per ottenere il meglio da lui dovete conoscere almeno l'essenziale della struttura degli oggetti Doopsi.

Iniziamo con una Scena:

```
nome interno

nome file dello sfondo

lista di Oggetti agganciati (nomi interni)
```

Il nome interno è un nome privato che l'utente attribuisce alla Scena ed è usato da Doopsi per riconoscere la Scena stessa. Il nome dello sfondo è il nome (completo di percorso) del file Iff contenente l'immagine da visualizzare. Segue poi una lista dei nomi interni degli Oggetti che devono essere posizionati sulla Scena.

Veniamo ora ad un Oggetto. Vi sono tre tipi di oggetti, ma tutti condividono la medesima struttura:

- a) gli Oggetti tout-court sono semplicemente oggetti nel senso usuale del termine e le loro immagini verranno disegnate sulla scena.
- b) gli AnimObject contengono un'animazione che verrà visualizzata sulla scena.
- c) le Zone Vuote sono costituite da un rettangolo (che non sarà disegnato) senza immagini.

Segue la struttura generale di un Oggetto:

```
nome

nome interno
```

coordinata X

coordinata Y

Larghezza

Altezza

stato

lista di immagini

nove programmi di azione

Il nome è ciò che appare sulla linea di testo del Player quando il puntatore è sull'oggetto. Il nome interno è un nome privato (possibilmente unico) che l'utente assegna all'oggetto ed è usato da Doopsi per riconoscere l'oggetto stesso: se l'utente utilizza più volte lo stesso nome interno l'Editor non si lamenta, ma potreste ottenere risultati imprevedibili (di solito Doopsi si ferma al primo oggetto che abbia il nome interno richiesto). Le coordinate X e Y specificano la posizione in cui disegnare l'oggetto sullo schermo. Larghezza ed Altezza sono le dimensioni dell'oggetto. Lo stato è un valore intero con nessun significato particolare: può essere usato per memorizzare lo stato Acceso-Spento di una lampadina, o lo stato Aperto-Chiuso di una porta, o il numero di monete che l'Omino possiede, e così via... Segue poi una lista di tutte le immagini (nome del file completo di percorso del file Iff contenente l'immagine) che l'utente ha agganciato all'Oggetto. Infine vi sono nove programmi (contenenti del codice Doopsi), uno per ciascun bottone della console del Player.

Un AnimObject differisce dalla precedente descrizione per il fatto che possiede anche una stringa di interi che specifica la sequenza in cui le immagini verranno visualizzate. Larghezza ed Altezza conterranno la massima larghezza ed altezza rispettivamente di tutte le immagini selezionate per l'animazione.

Una Zona Vuota differisce per il fatto che non verrà disegnata sulla scena, ma la sua posizione sarà comunque rivelata da Doopsi. La lista di immagini, però, non sarà necessariamente vuota in quanto l'utente potrebbe aver bisogno, ad esempio, di riporre questo oggetto nell'inventario (e quindi necessita di un'immagine di inventario), e così via.

1.37 DOOPSI-GS - Il Player

```
*****
Il Player.
*****
```

La sequenza di avviamento.

Quando lanciate il Player, esso cerca prima di tutto il file principale di Doopsi e, da questo momento in poi, tutte le azioni che intraprende sono determinate dai dati che trova in tale file. Dal file principale ricava il percorso dove cercare la "console" (la parte bassa dello schermo dove

appaiono i bottoni e l'inventario), contenuta nel file "panel.iff", e il file "player.dat". Se non trova uno qualunque dei pezzi che ho appena elencato si lamenta ed esce.

Supponendo che abbia svolto con successo i precedenti passi, il Player cerca ora nel file Doopsi (il file principale che contiene tutta la struttura del gioco) le animazioni del protagonista: devono esserci quattro animazioni, una per ciascuna direzione, con i nomi di default che l'utente ha scelto. Poiché questi dati sono vitali (come fate a giocare se non vedete dove siete?) il Player si ferma ed esce se non ha trovato tutto quello che stava cercando. Questo significa che esso deve trovare prima di tutto i quattro Oggetti che contengono tali animazioni e poi deve trovare anche tutte le immagini necessarie. A proposito, se l'utente fornisce anche le animazioni per far parlare il protagonista esse verranno caricate: in questo caso non è necessario fornire quattro animazioni se l'utente non lo ritiene necessario poiché tali animazioni vengono utilizzate solo se presenti. Ma ricordate, se l'utente chiede al Player di usare un'animazione questa animazione deve essere completa, vale a dire che il Player deve essere in grado di trovare tutte le immagini necessarie. Questa è una regola generale.

Ed ora veniamo al passo successivo: fino ad ora il Player è stato in grado di visualizzare la console, ma la metà superiore dello schermo è ancora terribilmente nera. Allora esso cerca nel file Doopsi la scena che avete scelto come prima scena usando l'editor di Preferenze: se l'utente si è dimenticato di includere almeno una scena con tale nome allora il Player non ha nient'altro da fare, perché non sa da quale scena iniziare il gioco, ed esce ("ancora!" potreste esclamare, ma non penso che questo comportamento sia dovuto a permalosità). Ovviamente esce anche se la scena è stata trovata ma non è presente il file Iff che contiene lo sfondo.

D'ora in poi il Player si lamenterà soltanto, senza uscire, se non troverà qualcosa. Errori comuni sono "Oggetto non trovato", se l'oggetto che l'utente richiede non è nella lista, o "File non trovato" se non trova il file contenente un'immagine o uno sfondo, ecc... In questi casi non succede nulla ed il gioco continua indisturbato. I messaggi di errore vengono visualizzati sulla console, nella stessa linea in cui appare la scritta "Vai verso"; per eliminare il messaggio e ritornare al gioco l'utente deve premere il tasto destro del mouse.

Se una scena è stata caricata con successo lo sfondo verrà visualizzato e tutti gli oggetti in essa contenuti saranno disegnati sullo schermo: gli AnimObject sono automaticamente riconosciuti ed animati. Supponendo che sappiate già (dal manuale dell'Editor) cos'è un percorso Doopsi, vi dico che il Player cerca prima un nodo nel percorso della scena corrente con il nome uguale a quello della scena da cui proviene, poi, se questo fallisce, cerca un nodo che ha lo stesso nome della scena corrente; infine disegna l'Omino (che guarda l'utente) vicino al nodo trovato.

C'è anche qualcosa di leggermente tecnico che dovete sapere a proposito dei nodi: questi vi permettono di fornire una animazione diversa da quella di default per l'Omino e di decidere dove cambiare scena; inoltre il percorso della scena può essere influenzato da condizioni contenute nei suoi nodi. Il Player legge il nome delle animazioni "custom", le condizioni ed i cambi di scena da un nodo quando l'Omino ha i piedi sopra quel nodo: l'animazione custom viene visualizzata solo quando l'Omino cammina dal nodo attuale a quello successivo (così, se l'Omino si ferma sul nodo attuale voi

non potete vedere tale animazione e dovete fornire un fotogramma di riposo all'animazione precedente), ma le condizioni e i cambi di scena sono letti ed eseguiti immediatamente quando l'Omino arriva sul noto attuale. Per riassumere: le animazioni custom di un nodo vengono eseguite quando l'Omino si allontana da quel nodo, ma le condizioni e i cambi di scena vengono eseguiti quando l'Omino arriva a quel nodo.

Alla fine il gioco inizia e voi diventate i protagonisti.

Errori.

Segue una breve descrizione dei possibili errori. I messaggi di errore compaiono nella riga di testo e vengono cancellati premendo il tasto destro del mouse. Se l'errore è del tipo "... non trovato ..." il Player stampa anche il nome dell'oggetto (o del file, ecc.) che non ha trovato.

Oggetto non trovato:	l'oggetto non è nella lista principale.
Scena non trovata:	la scena che avete richiesto non è nella lista.
File non trovato:	non penso ci sia bisogno di spiegazioni per questo.
Spot non trovato:	l'utente ha richiesto uno spot inesistente.
Inventario pieno:	la lista dell'inventario è capiente ma limitata, quindi se l'utente vi ripone troppi oggetti essa può traboccare.
Niente spazio per l'animazione:	il numero di AnimObject che il Player può gestire contemporaneamente è limitato.
Dialogo non trovato:	l'utente ha richiesto una pagina di dialogo che non esiste.
Directory non trovata:	questo messaggio compare nel file lister: l'utente ha specificato un percorso inesistente.
Niente spazio per l'oggetto:	il numero di oggetti che una scena può contenere è limitato.
Profondità eccessiva:	l'utente ha tentato di disegnare sulla scena un oggetto con profondità maggiore di quella della scena stessa.
EndIf senza If:	in un programma Doopsi vi è un'istruzione EndIf senza il corrispondente comando If.
File Iff non valido:	dovete fornire un file Iff standard quando esplicitamente richiesto, ad es.: nelle istruzioni ShowPicture e SoundPlay.
Stack pieno:	si possono nidificare fino a 10 sottoprogrammi: se ne aggiungete uno di troppo lo stack si esaurisce.

1.38 DOOPSI GS - Player Royalty

```
*****
Player royalty.
*****
```

Il Player di Doopsi può essere liberamente distribuito insieme alle vostre avventure, ma non a scopo di lucro.

Se lo includete in un pacchetto Shareware dovete spedirci una versione

registrata dello pacchetto stesso.

Se lo includete in un programma commerciale dovete pagarci la somma di US\$50 (Lit. 50000) e dovete chiederci un'autorizzazione scritta.

Per maggiori informazioni contattate:

Fabio Rotondo e-mail: fsoft@intercom.it

1.39 DOOPSI-GS - Limiti

Limiti di Doopsi.

Qui sono riportate le limitazioni attuali di Doopsi: le diciture "Shareware" e "Special" indicano le limitazioni per le due versioni dimostrative.

Editor:

Tipo ~~~~	numero Max ~~~~~	
scene	200	(Shareware: 3; Special: 5)
oggetti	500	(Shareware: 30; Special: 40)
pagine di dialogo	100	(Shareware: 5; Special: 8)
linee per pagina di dialogo	12	(Shareware: 3; Special: 4)
immagini per oggetto	15	
oggetti sulla scena contemporaneamente	20	
lunghezza nomi interni	30	

Player: gli stessi dell'Editor più

Tipo ~~~~	numero Max ~~~~~	
animobject sulla scena contemporaneamente	10	
oggetti in inventario	500	(Shareware: 30; Special: 40)
sottoprogrammi nidificati	10	

gruppi If nidificati 10

1.40 DOOPSI-GS - Shareware

REGISTRATION FORM

Utilizzate la parte seguente di questo documento per registrarvi a DOOPSI-GS.

Attualmente le modalità di pagamento sono le seguenti:

Moneta sonante, vaglia postali e basta.

Ci occuperemo della spedizione della vostra versione registrata di DOOPSI non appena avremo la certezza che il versamento sia stato fatto. Nel caso di un vaglia postale, potete accelerare i tempi spedendoci l'attestato di versamento, unitamente al modulo sottostante compilato in ogni sua parte.

Insieme alla versione registrata di DOOPSI-GS riceverete la documentazione completa nei formati elettronici da voi richiesti.

----- TAGLIA QUI ----- TAGLIA QUI ----- TAGLIA QUI -----

FOGLIO DI REGISTRAZIONE.

Vorrei registrarmi al vostro programma.
Vi spedisco l'ammontare di Lit. 35000

Segnare se necessario:

☐ più le spese postali di Lit. 3000 perché voglio il dischetto via
Posta.

☐ più Lit. 10000 perché voglio ricevere via posta i prossimi 3 upgrade.
(NB. Nel caso possiate ricevere via e-mail gli upgrades, questi sono
GRATUITI)

☐ voglio ricevere informazioni su quanto costa il manuale in formato
cartaceo.

Versione Programma: Doopsi-GS V1.00

Nome :

E-Mail :

Indirizzo :

Telefono :

Computer :

Vorrei avere la documentazione elettronica gratuita nei seguenti formati
(segnare):

☐ PostScript.

☐ LaTeX.

----- TAGLIA QUI ----- TAGLIA QUI ----- TAGLIA QUI -----

Spedire a:

Fabio Rotondo
c.so Vercelli, 9
28100 Novara
Italia
E-Mail: fsoft@intercom.it

Andrea Galimberti
via Villoresi, 87
20029 Turbigo (Mi)
Italia