

- Manuale dell'Editor

Sommario

1. Introduzione
 - ◆ 1.1. Informazioni generali
 - 1.2. Caratteristiche non ancora implementate
2. Creazione di una mappa
 - ◆ 2.1. Creare una nuova mappa
 - 2.2. Caricamento/Salvataggio
3. Modifica di una mappa
 - ◆ 3.1. Terreno e oggetti
 - 3.2. Modificare oggetti
 - ◇ 3.2.1. Città
 - 3.2.2. Rovine e templi
 - 3.3.3. Armate
4. Altre caratteristiche
5. Gestione dei giocatori
6. Gestione degli eventi
 - ◆ 6.1. Introduzione al sistema degli eventi
 - 6.2. Come modificare il sistema degli eventi
 - 6.3. Eventi
 - 6.4. Reazioni
 - 6.5. Condizioni
 - 6.6. Alcuni esempi
7. Come modificare lo scenario manualmente

1. Introduzione

1.1. Informazioni generali

L'editor delle mappe è stato aggiunto nella versione 0.3.5. Anche se ancora piuttosto incompleto e rudimentale, spero sia comunque abbastanza sviluppato da darvi la possibilità di divertirvi un po' creando nuove mappe. In ogni caso, non esitate a contattarmi per segnalare bug o nuove funzionalità da implementare. Il manuale verrà aggiornato prima di ogni nuova release.

1.2. Caratteristiche non ancora implementate

Dopo aver lavorato ininterrottamente al codice per gli ultimi tre mesi ho cominciato ad essere piuttosto stanco, perciò siate comprensivi se notate delle caratteristiche non ancora implementate. Provvederò nelle versioni future. Per il momento vi rimando alla sezione 7 del manuale, che contiene alcune informazioni e suggerimenti, per sopperire alle momentanee mancanze dell'editor.

Caratteristiche da migliorare o non ancora implementate:

- mancano i rapporti su templi e rovine
- manca il supporto per la tastiera
- il layout e la maneggevolezza degli strumenti di lavoro non sono perfetti
- il sistema degli eventi è piuttosto complicato (e probabilmente non verrà modificato in tempi brevi)

Funzionalità da implementare:

- non è ancora possibile impostare i punti ferita di un'armata a un livello inferiore al massimo (all'inizio del gioco tutte le armate hanno il massimo dei punti ferita). Lo stesso vale per i punti movimento.
- non è possibile posizionare manufatti sulla mappa
- non è possibile assegnare agli eroi una quest iniziale
- gli oggetti non possono essere spostati, ma solo eliminati o creati ex-novo
- manca il supporto della versione (se la versione del gioco salvato viene modificata, sarà necessario applicare manualmente una patch al file dello scenario)

2. Creazione di una mappa

2.1. Creare una nuova mappa

Selezionate "Nuova Mappa" nel menu "File". Viene visualizzata una finestra di dialogo in cui potete selezionare il tipo di terreno (al momento è disponibile solo quello di default), le dimensioni e lo stile di riempimento, ovvero il tipo di terreno da utilizzare per la riempitura iniziale della mappa. "Mappa casuale" è uno stile di riempimento particolare, che consente di generare una mappa casuale, completa di città, templi, rovine, etc. Nella finestra successiva potrete impostare la distribuzione del terreno e il numero di siti (città, rovine e templi), ai quali viene assegnato un nome predefinito.

In alternativa, potete caricare il file "random.map" contenuto nella directory di salvataggio di Freelords (in genere \$HOME/.freelords). Il file non è altro che l'ultima partita casuale giocata e contiene tutti i dati relativi ai giocatori, ai siti e agli eventi standard.

2.2. Caricamento/Salvataggio

Il menu "File" contiene i comandi di caricamento e salvataggio. Per caricare una mappa, dopo aver scelto "Carica" viene visualizzata una finestra che vi consente di selezionare un file mappa con estensione ".map".

Quando selezionate il comando "Salva", la mappa viene salvata utilizzando il nome file più recente (ad esempio quello del file dal quale avete caricato la mappa oppure quello dell'ultimo file salvato). Nel caso in cui non esista un nome file utilizzabile, il comando "Salva" assume le stesse funzioni del comando "Salva con nome", visualizzando una finestra che vi consente di indicare il nome del file e la directory in cui salvarlo. L'estensione ".map" viene aggiunta automaticamente nel caso in cui non sia stata specificata.

3. Modifica di una mappa

3.1. Terreno e oggetti

Dopo aver caricato o creato una nuova mappa, nella parte destra dell'editor sono disponibili diversi pulsanti che consentono di modificare il terreno e di aggiungere siti e armate alla mappa.

Potete selezionare il tipo di terreno da utilizzare facendo clic sulle varie icone dei terreni o sui pulsanti specifici che consentono di:

- modificare il terreno in un'area di 1x1
- modificare il terreno in un'area di 3x3
- rimuovere oggetti (pulsante contrassegnato da una croce di colore rosso). Tenete presente che l'editor non chiederà conferma prima della rimozione e che i primi oggetti ad essere rimossi saranno le armate.
- posizionare armate, città, rovine e templi

Per la maggior parte degli oggetti esistono dei vincoli di posizionamento. Ad esempio, se tentate di posizionare una città sull'acqua, il terreno verrà automaticamente modificato in erba. Ciò non vale però per le armate, quindi, quando posizionate delle armate sull'acqua, assicuratevi che tali armate siano composte da unità in grado di spostarsi sull'acqua.

3.2. Modificare oggetti

In generale, per modificare un oggetto potete fare clic su di esso con il pulsante destro del mouse. Nel caso in cui più oggetti siano raggruppati insieme (ad esempio, un'armata in una rovina), vi verrà chiesto di indicare l'oggetto da modificare.

3.2.1. Città

Facendo clic con il pulsante destro del mouse su una città viene visualizzata la finestra relativa. Le operazioni di modifica sono molto intuitive. È però utile approfondire alcuni concetti.

La differenza tra produzione base e produzione avanzata è stata già illustrata nel manuale (base = set di unità di default uguale per ogni giocatore; avanzata = set di unità specifico per ogni giocatore, perso quando una città viene conquistata).

La finestra di modifica contiene però anche un menu a discesa che consente di scegliere tra "Intatta" e "Distrutta". Quando un giocatore occupa una città, può scegliere di distruggerla, radendo quindi al suolo l'intera città e riducendola a una rovina. Se desiderate posizionare sulla mappa una città già distrutta fin dall'inizio, selezionate "Distrutta".

3.2.2. Rovine e templi

La principale modifica che è possibile apportare riguarda il nome. Nel caso delle rovine, potete modificare anche l'occupante (con il quale l'eroe dovrà combattere per entrare nella rovina) facendo clic sul pulsante "Guardiano". La finestra che verrà visualizzata è identica a quella relativa alle armate.

3.2.3. Armate

Facendo clic con il pulsante destro del mouse su un'armata viene aperta una finestra di dialogo piuttosto intuitiva.

Potete modificare il proprietario di un'armata, ma assicuratevi che la modifica sia coerente. Ad esempio, fate attenzione a non posizionare in una città neutrale un'armata appartenente a un giocatore (del resto, la creazione di una mappa non serve a evidenziare i bug del gioco, no? :)).

Potete modificare anche le proprietà delle singole unità che compongono un'armata. Questa possibilità è particolarmente utile per regolare con più verosimiglianza la forza di un'armata, ad esempio per creare un'armata di veterani o un'armata piuttosto malandata.

4. Altre caratteristiche

Dal menu è possibile accedere anche a:

- una finestra di dialogo "Trova", che consente di ricercare un oggetto (città, rovina, tempio, armata o unità) immettendone l'ID. Quando premete Invio, la mappa verrà centrata sulla posizione dell'oggetto.
- rapporti su armate e città. Fate clic su uno degli oggetti per centrare la mappa su di esso.

La gestione dei giocatori e degli eventi viene trattata nelle sezioni seguenti.

5. Gestione dei giocatori

Quando selezionate "Giocatori" nel menu "Modifica", viene visualizzata la finestra per la gestione dei giocatori.

Cliccando su "Precedente" e "Successivo", è possibile selezionare tutti i giocatori. Potete usare i due menu a scomparsa per selezionare il tipo di giocatore e quello delle sue armate. Ci sono quattro tipi di giocatori (come nel gioco): Umano, intelligenza artificiale semplice (non fa niente), intelligenza artificiale veloce (assembla le armate e attacca) e l'intelligenza artificiale avanzata (cerca di proteggere le città, ecc..).

I campi editabili sottostanti dovrebbero essere auto esplicanti; è possibile impostare il colore, il nome e l'ammontare iniziale di oro per il giocatore.

È sempre presente un giocatore neutrale che non è possibile rimuovere (e non dovrebbe essere rimosso). In aggiunta, è possibile creare giocatori persistenti (immortali e non muoiono quando perdono tutte le città). Nel caso speciale in cui viene selezionata l'intelligenza artificiale veloce, ci sono due opzioni aggiuntive che possono essere scelte. La prima permette al giocatore di unire due armate vicine per aumentarne la forza. La seconda permette di avere un'intelligenza artificiale maniaca. In questo caso, attaccherà in maniera decisamente casuale (interessante per avere ad esempio mostri erranti).

Quando aggiungete un nuovo giocatore, vi sarà chiesto se aggiungerlo prima o dopo il giocatore attuale (questo per avere un controllo sulla posizione nel round).

6. Gestione degli eventi

6.1. Introduzione al sistema degli eventi

In genere, quando si crea uno scenario, lo si arricchisce di caratteristiche che lo rendono più realistico, come ad esempio la possibilità di inserire delle imboscate o di aggiungere nuovi obiettivi, quali esplorare una rovina in particolare o raggiungere un luogo preciso con una specifica armata. In FreeLords, ciò può essere fatto attraverso il sistema di eventi integrato: un sistema molto potente, ma anche piuttosto complesso e il cui utilizzo può dover richiedere anche una certa capacità di programmazione.

A livello generale, il sistema degli eventi è composto da tre elementi base: eventi, reazioni e condizioni.

- Gli eventi vengono attivati da alcune azioni particolari. Ad esempio, l'evento `RuinSearch` viene attivato quando un giocatore esplora una rovina specifica.
- Ad ogni evento possono essere collegate una o più reazioni. Quando viene attivato un evento, vengono attivate anche tutte le reazioni ad esso collegate. Tali reazioni possono essere, ad esempio, "Uccidi un giocatore specifico" oppure "Vinci la partita".
- Ogni evento e ogni reazione possono avere una o più condizioni. In casi del genere, l'evento o la reazione vengono attivati soltanto se la condizione è vera. Ad esempio, per assicurarsi che l'evento `CityConquered` sia attivato solo se la città viene conquistata da un determinato giocatore, è possibile assegnare all'evento una condizione che verifichi che il giocatore attivo in quel momento coincida con il giocatore scelto per l'attivazione dell'evento.

Entrando nel dettaglio...

Agli eventi sono associati un id univoco (a cui si può dover fare riferimento), una serie di condizioni e reazioni (a cui si è già accennato), alcuni dati specifici relativi al tipo di evento e una stringa di commento. Quest'ultima non ha effetti sul gioco, ma viene utilizzata per fornire spiegazioni sull'evento e sulle sue finalità. All'inizio del gioco, ciascun evento può essere abilitato o disabilitato. Dopo che un'evento è stato attivato (cosa che può avvenire solo se è abilitato), l'evento si disabilita automaticamente. Per tutta la durata della partita gli eventi non vengono mai eliminati: è perciò possibile (ri)abilitare un evento in qualsiasi momento.

Le Condizioni hanno uno o più elementi di informazione (es. l'id del giocatore che deve essere attivo). Nel caso in cui un evento o una reazione siano collegati a una condizione, prima di attivare l'evento o la reazione il sistema esegue un controllo sulla condizione. Solo se la condizione è verificata l'evento o la reazione vengono attivati. Tenete comunque presente che l'evento rimane abilitato anche nel caso in cui la condizione non sia verificata.

Le reazioni, infine, possono essere associate a diverse condizioni e a una serie di elementi di informazione. È importante notare che le reazioni associate agli eventi vengono attivate nell'ordine in cui sono state aggiunte. Tenetelo presente nel caso in cui, ad esempio, desiderate posizionare un'unità, visualizzare un messaggio e quindi rimuovere l'unità.

Gran parte degli eventi, delle reazioni e delle condizioni utilizzano come parametro l'id di un oggetto. In questi casi, se tale id non esiste, l'evento, la reazione o la condizione non verranno attivati. Ciò può creare problemi nel caso delle condizioni: alcune di esse potrebbero ritornare sempre un valore positivo, mentre altre negativo.

6.2. Come modificare il sistema degli eventi

Per modificare eventi, condizioni e reazioni esistono tre finestre di dialogo specifiche, ognuna delle quali ha caratteristiche particolari.

Quando aggiungete un evento o una reazione, viene visualizzato un elenco da cui è possibile selezionare il tipo di evento o reazione desiderato. Tenete presente che, una volta eseguita una selezione, la scelta non può più essere modificata a meno che non rimuoviate prima l'evento o la reazione selezionata.

La finestra degli eventi contiene tutti gli eventi disponibili e consente di selezionare un evento, modificarne le proprietà, etc. Tramite i pulsanti, potete aggiungere reazioni e condizioni, nonché aggiungere o rimuovere eventi. I nuovi eventi vengono sempre visualizzati in fondo all'elenco.

Quando fate clic sul pulsante "Reazioni", viene visualizzata una finestra di dialogo che, per molti versi, è simile a quella degli eventi (i diversi campi vi consentiranno di modificare i valori delle reazioni, associare condizioni, etc.). Tuttavia, dal momento che l'ordine delle reazioni è importante (le prime reazioni saranno anche le prime ad essere attivate), la procedura da utilizzare per aggiungere le reazioni è leggermente differente da quella relativa agli eventi. Quando volete aggiungere una nuova reazione, selezionate dalla lista delle reazioni già aggiunte quella che dovrà seguire la nuova e fate clic su "Aggiungi". La nuova reazione verrà quindi aggiunta prima della reazione selezionata. Per aggiungere una reazione in fondo all'elenco, fate clic su "---fine---" in fondo alla lista e quindi fate clic su "Aggiungi".

Anche la finestra delle condizioni ha alcune caratteristiche particolari. Quando aggiungete una condizione, in fondo all'elenco viene aggiunta automaticamente una condizione standard. Il tipo di questa condizione può essere modificato in qualsiasi momento utilizzando il menu a scorrimento a destra.

6.3. Eventi

Attualmente sono previsti 11 eventi diversi:

KillAll	Viene attivato quando tutti i giocatori hanno perso tranne uno. Il giocatore neutrale non viene considerato (quindi non ha importanza risulti già sconfitto o meno)
PlayerDead	Viene attivato quando un giocatore specifico viene sconfitto. Come parametro viene utilizzato l'id del giocatore.
CityConq	Viene attivato quando una città è conquistata o razziata. Il parametro utilizzato è l'id della città. Tenete presente che una volta razziata una città non può più essere conquistata e che ciò può accadere spesso, perciò assicuratevi che l'evento non sia ignorato.
ArmyKilled	Si attiva quando un'unità viene sconfitta. Il parametro utilizzato è l'id dell'unità.
Round	Viene attivato all'inizio di un turno specifico. Il parametro utilizzato è il numero del round. Se collegato alla condizione Player potete usare questo evento per attivare una reazione all'inizio del turno di un giocatore specifico.
RuinSearch	Si attiva quando viene esplorata una rovina. Il parametro utilizzato è l'id della rovina.
TempleSearch	Equivalente a RuinSearch, ma riferito ai templi.
Dummy	Questo evento non viene mai attivato, ma può risultare utile, in particolare, se collegato alla reazione RaiseEvent e alla condizione Counter. Fate riferimento all'esempio nella sezione 6.6.
NextTurn	È simile all'evento Round, ma viene attivato successivamente. Se collegato alla condizione Player e alla reazione ActEvent, può essere utilizzato da altri eventi per segnalare a un giocatore specifico che qualcosa è avvenuto nel momento in cui altri eventi sono stati attivati. Per ulteriori informazioni, fate riferimento alla sezione 6.6.
StackKilled	È simile ad ArmyKilled, ma si attiva quando viene sconfitta un'intera armata. Di per sé, è un evento superfluo (può essere emulato utilizzando gli eventi ArmyKilled o Dummy oppure una condizione Counter e una reazione RaiseEvent), ma è molto comodo come scorciatoia. Quando lo utilizzate, tenete presente che più armate possono essere unite insieme (quando si uniscono due armate una delle due viene considerata come distrutta) e che le unità che

compongono un'armata possono venire aumentate o divise in più armate. In caso di dubbio, potete comunque sempre utilizzare l'evento ArmyKilled. Il parametro utilizzato da StackKilled è l'id dell'armata.

StackMove Questo evento utilizza come parametro una posizione e viene attivato quando un'armata raggiunge la posizione indicata.

6.4. Reazioni

Al momento, esistono 15 reazioni:

Message	Visualizza un messaggio di testo in una finestra. Il testo è anche il parametro della reazione.
AddGold	Aumenta l'oro posseduto da un giocatore. Utilizza due parametri: l'id del giocatore e la quantità di oro. Se specificate una quantità negativa, l'oro del giocatore diminuisce.
AddUnit	Aggiunge un'armata a quelle possedute da un giocatore. Utilizza due parametri: l'id del giocatore e la posizione in cui verrà visualizzata l'armata. Per scegliere il tipo di armata, fate clic sul pulsante "Armata" che viene visualizzato quando selezionate questa reazione. Se la posizione è impraticabile, l'armata verrà posizionata in una delle caselle adiacenti alla posizione specificata. Nel caso ciò non sia possibile, la reazione non avrà esito (potrebbe essere considerato un bug?).
DelUnit	Elimina un'armata specifica. Il parametro utilizzato è l'id dell'armata.
Update	Questa reazione non utilizza nessun parametro. Il suo unico effetto è quello di aggiornare la schermata del gioco. Ciò può essere utile, ad esempio, se volete aggiungere un'unità e poi visualizzare un messaggio. Normalmente, la schermata non verrebbe aggiornata immediatamente quindi non riuscireste a vedere l'unità. Per visualizzarla, utilizzate questa reazione dopo l'aggiunta dell'unità e prima la visualizzazione del messaggio.
Center	Centra lo schermo su un determinato punto della mappa, aggiornando la visualizzazione durante il processo.
CenterObject	È equivalente a Center, ma centra lo schermo su un oggetto, che può essere una città, una rovina, un tempio, un'armata o un'unità. È particolarmente utile per spostare oggetti (armate o, più frequentemente, unità).
CreateItem	Crea un manufatto in una posizione specifica. I parametri utilizzati sono le coordinate x/y del punto in cui volete visualizzare il manufatto e l'indice dell'elemento stesso. Poiché al momento l'editor non dispone di un visualizzatore di manufatti, vi consiglio di fare riferimento al file items.xml, dove potrete trovare gli indici dei manufatti tra i tag <d_index> e </d_index>.
WinGame	Termina il gioco con la vittoria. Come parametro utilizza un numero. Attualmente questa reazione non ha alcuna utilità, ma verrà probabilmente utilizzata nelle versioni future in caso di campagne non lineari. Per il momento, ignoratela.
LoseGame	Equivalente a WinGame. L'unica differenza è la schermata che viene visualizzata, che in questo caso fa riferimento a una sconfitta piuttosto che a una vittoria. Ad ogni modo, al momento la schermata in questione non è stata ancora implementata (ma lo sarà presto 😊)
RaiseEvent	Questa reazione consente di attivare un altro evento. Per ulteriori informazioni, fate riferimento alla sezione 6.6. Il parametro utilizzato è l'id dell'evento da attivare.
ActEvent	Abilita o disabilita un evento. Utilizza due parametri: l'id dell'evento e il valore 0 (falso) o 1 (vero) che determina se l'evento verrà abilitato (1) o disabilitato (0). Può essere utilizzata, ad esempio, nel caso di eventi persistenti. Normalmente, infatti, gli eventi vengono disabilitati dopo essere stati attivati: la reazione ActEvent, se aggiunta alla catena che riabilita l'evento, consente di attivare l'evento infinite volte.
RevivePlayer	Rimette in gioco un giocatore sconfitto. L'argomento utilizzato è l'id del giocatore. Tenete

presente che al giocatore riportato in vita dovrà essere assegnata anche una città (reazione TransferCity), altrimenti verrà sconfitto di nuovo in pochissimo tempo.

KillPlayer Elimina un giocatore. Il parametro utilizzato è l'id del giocatore. Tutte le armate possedute dal giocatore eliminato verranno rimosse, e tutte le sue città assegnate al giocatore neutrale.

TransferCity Trasferisce la proprietà di una città a un giocatore specifico. I parametri da specificare sono l'id del giocatore a cui deve essere assegnata la città e l'id della città stessa.

6.5. Condizioni

Attualmente esistono 5 condizioni:

Player Verifica che un giocatore specifico sia effettivamente attivo. È particolarmente utile se connessa all'evento Round, In questo modo, l'evento verrà attivato solo quando inizia il turno del giocatore in questione. Il parametro utilizzato è l'id del giocatore.

Counter È una condizione particolare: utilizza infatti come parametro un contatore il cui valore decresce ogni volta che viene attivato l'evento o la reazione collegati ad essa. Quando il contatore raggiunge lo zero, la condizione esaurisce la sua funzione. Per un esempio su come utilizzare questa condizione, fate riferimento alla sezione 6.6.

Living Verifica che un determinato giocatore sia ancora in gioco. Il parametro utilizzato è l'id del giocatore.

Dead Verifica che un determinato giocatore sia stato sconfitto. Il parametro utilizzato è l'id del giocatore.

Army Immettendo come parametro l'id di un'unità, la condizione verifica che tale unità faccia parte dell'armata correntemente attiva (per armata attiva si intende un'armata che sta compiendo una qualche azione, ad esempio l'esplorazione di una rovina). Se collegata all'evento StackMove, questa condizione può essere utile per determinare se quell'unità si è spostata in una posizione specifica.

6.6. Alcuni esempi

a) Impostazioni standard

- Primo esempio: impostazioni standard per una partita qualsiasi a cui prendono parte alcuni giocatori umani e altri controllati dalla cpu.

Obiettivo: quando un giocatore viene sconfitto, tutti gli altri giocatori (umani) devono esserne informati. Se un giocatore umano sconfigge tutti gli avversari allora la partita finisce ed è vinta da quel giocatore, altrimenti la partita è persa.

Realizzazione: Mettendo in conto la sconfitta di un qualsiasi giocatore X, create un evento NextTurn (disabilitato) per ognuno degli altri giocatori umani rimasti (eccetto X nel caso sia anch'esso umano) collegato a una reazione Message che visualizzi il messaggio "Il giocatore X è stato sconfitto". Create quindi un evento PlayerDead con l'id di X, collegandolo a reazioni che abilitino ognuno degli eventi NextTurn. In questo modo, quando X viene sconfitto, ognuno degli altri giocatori umani rimasti riceve un messaggio all'inizio del proprio turno che lo informa della sconfitta di X.

Infine, create un evento KillAll e collegatelo, per ciascun giocatore X ad eccezione di quello neutrale, sia con una reazione Message che con una reazione WinGame (se X è umano) o LoseGame (se X è un'intelligenza artificiale). A entrambe le reazioni è collegata una condizione Living a cui è associato l'id del giocatore X.

Nota: Se create uno scenario senza eventi, la partita non avrà mai fine! Un numero minimo di eventi è necessario, almeno quelli che regolano fattori quali vincere o perdere una partita.

b) utilizzo dell'evento Dummy e della condizione Counter

- In alcune situazioni la condizione Counter può risultare particolarmente utile. Prendete il caso, ad esempio, di uno scenario con tre rovine. Se volessimo far vincere un giocatore soltanto dopo aver esplorato tutte e tre le rovine, come dovremmo procedere?

La risposta è semplice. Create un evento Dummy collegato a una reazione WinGame e a una condizione Counter. Impostate il contatore della condizione su tre. Ora, per ogni rovina, create un evento RuinSearch (con la condizione che il giocatore in questione sia attivo) collegato a una reazione RaiseEvent alla quale dovrete assegnare l'id dell'evento Dummy.

In questo modo, ogni volta che il giocatore in questione esplora una rovina, viene attivato uno dei tre eventi RuinSearch, la cui reazione, a sua volta attiva l'evento Dummy. Poiché una rovina può essere esplorata una sola volta, il contatore decresce di una unità ad ogni rovina esplorata. Quando tutte e tre le rovine saranno state esplorate il contatore raggiungerà lo zero e la partita sarà vinta.

Tenete presente che in questo esempio non viene considerato il caso in cui ci siano altri giocatori che possano esplorare le rovine. In tal caso, infatti, la partita non avrebbe mai fine, a meno che non utilizziate un numero maggiore di eventi.

7. Come modificare lo scenario manualmente

Se desiderate arricchire lo scenario con qualcosa di insolito che non sia stato ancora implementato, avete a disposizione due possibilità.

La prima consiste nell'inviare un'email al nostro elenco degli sviluppatori. Poiché ad occuparsi delle mappe non saranno in molti, avere qualcun altro disposto a collaborare consentirà (oltre a dotare FreeLords di nuove mappe) di velocizzare i tempi di implementazione delle nuove caratteristiche, cosicché, per vedere le modifiche all'opera, sarà sufficiente aspettare una nuova release o versione cvs.

In alternativa, se proprio non potete aspettare o se avete bisogno, ad esempio, di aggiornare il file dello scenario a una nuova versione, l'unica possibilità che avete è quella di modificare il file manualmente. Non scoraggiatevi! È più facile di quello che può sembrare.

Il file dello scenario utilizza un formato XML specifico. Vi accorgerete che, semplicemente aprendo il file con un editor di testo, sarete probabilmente già in grado di identificarne delle sezioni. Dovrete comunque tenere presenti almeno tre cose in particolare (a parte ricordarvi di fare una copia di backup!!!):

1. I dati relativi ad ogni oggetto del gioco (armate, città, etc.) sono contenuti tra due tag. Ad esempio, i dati di un giocatore sono compresi tra <player> e </player>. Anche tutti i sotto-oggetti sono compresi tra tag. Ad esempio, i dati di un'armata composta da più unità avranno l'aspetto seguente:

<stack>

◆ ..dati armata... <army>

◇ ..dati prima unità...

</army> <army>

◇ ..dati seconda unità...

```
</army>
♦ ..
</stack>
```

I dati veri e propri sono compresi tra tag che iniziano con "d_". Ecco un esempio di tag relativi a un giocatore:

```
<player>
    ♦ <d_id>4</d_id> <d_name>Genio del male</d_name>
    ♦ ..
</player>
```

Il grosso delle cose da sapere per eseguire modifiche manuali è tutto qua.

2. Il significato di ciascun tag è descritto nel file Savefile all'interno della directory della documentazione.
3. Il numero della versione è contenuto nel primo tag di apertura del file dello scenario. È questo il numero che dovete modificare quando cambiate la versione per far sì che il parser accetti la vostra mappa. Assicuratevi, tuttavia, di leggere e applicare le modifiche elencate nel file Savefile relative al nuovo numero di versione (o ai numeri precedenti nel caso saltiate qualche numero), altrimenti la mappa potrebbe non funzionare. Se volete inviarci una vostra mappa e avete dei problemi riguardo la versione, siamo pronti a fornirvi tutta l'assistenza necessaria.

Tutto qua. Se non siete del tutto soddisfatti di questo manuale, ricordate che l'Open Source vive di singoli contributi, e il contributo minimo, in questo caso, è quello di contattarci! *HintHintHint* 😊

Manuale dell'Editor (last edited 2007-04-16 11:18:55 by [ulf](#))