

- Editoranleitung

Inhalt:

1. Einführung

- 1.1. Über dieses Dokument
- 1.2. Nicht implementierte Sachen

2. Erzeugen einer Karte

- 2.1. Erzeugen einer neuen Karte
- 2.2. Laden/Speichern

3. Editieren der Karte

- 3.1. Plazieren von Terrain und Objekten
- 3.2. Ändern von Objekteigenschaften
 - ◆ 3.2.1. Städte
 - 3.2.2. Ruinen und Tempel
 - 3.2.3. Armeen

4. Einige andere Menüeinträge

5. Verwalten von Spielern

6. Verwalten von Ereignissen

- 6.1. Einführung in das Ereignissystem
- 6.2. Editieren von Ereignissen
- 6.3. Liste aller Ereignisse
- 6.4. Liste aller Reaktionen
- 6.5. Liste aller Bedingungen
- 6.6. Einige Beispiele

7. Manuelles Editieren eines Szenarios

1. Einführung

1.1. Über dieses Dokument

Diese Anleitung soll einigermaßen vollständig den Karteneditor beschreiben. Der Editor ist seit Version 0.3.5. Bestandteil von FreeLords, und zur Zeit etwas unvollständig und unbequem zu bedienen. Ich hoffe, daß es trotzdem Spaß macht, mit dem Editor Karten zu erzeugen. Falls Bugs auftreten oder ein Feature wichtig ist und fehlt, schreib einfach eine Mail. Möglicherweise ist das Dokument auch nicht ganz aktuell, auch wenn es vor Releases regelmäßig aktualisiert werden sollte.

1.2. Nicht implementierte Sachen

Ich habe den Editor jetzt seit drei Monaten programmiert und kann den Quellcode mittlerweile nicht mehr sehen. Ich hoffe, daß es nicht zu furchtbar ist, wenn einige Funktionen fehlen. Das wird in der Folgezeit nachgeholt. Bis dahin gibt Abschnitt 7 einige Tips, wie man Sachen machen kann, die der Editor nicht beherrscht.

Grundsätzlich ist folgendes nicht implementiert:

- Auflistungen von Tempeln und Ruinen fehlen
- Tastaturunterstützung fehlt
- Layout und Nutzbarkeit sind manchmal unter aller Sau
- das Ereignissystem ist ziemlich kompliziert (das liegt allerdings am Ereignissystem und wird sich kaum ändern)

Fehlende Funktionen sind:

- Man kann die tatsächlichen Lebenspunkte einer Einheit nicht einstellen, nur die maximalen Lebenspunkte (alle Einheiten sind zu Beginn des Spiels kerngesund); dasselbe gilt für Bewegungspunkte
- Man kann keine Artefakte auf der Karte platzieren
- Helden können keine Missionen zugewiesen werden
- Es ist nicht möglich, Objekte zu bewegen, man kann sie nur Löschen und woanders platzieren
- Versionsunterstützung; wenn sich die Version der Savegames ändert, muß man das Szenario manuell editieren

2. Erzeugen einer Karte

2.1. Erzeugen einer neuen Karte

Im "Datei"-Menü wählt man dazu "Neue Karte" an. Im ersten Dialog kann man die Geländeart (zur Zeit existiert nur der Standardsatz), die Größe und den Terraintyp auswählen. Mit dem Terraintyp wählt man aus, mit welchem Terrain die Karte anfangs gefüllt werden soll. Ein spezieller Terraintyp ist "Zufallskarte". Dabei wird eine zufällige Karte mit Städten, Tempeln und Ruinen erzeugt. Die Terrainverteilung und die Anzahl der einzelnen Gebäude kann dann im folgenden Dialog eingestellt werden. Allerdings haben die Gebäude keine richtigen Namen.

Alternativ kann man auch die Datei "random.map" im Spielstandsverzeichnis von FreeLords (üblicherweise \$HOME/.freelords) laden. In dieser Karte sind bereits die Spieler initialisiert, die Gebäude haben vernünftige Namen und ein paar Standardereignisse sind gesetzt. Diese Datei beinhaltet die zuletzt gespielte Zufallskarte.

2.2. Laden/Speichern

Das "Datei"-Menü beinhaltet auch Einträge zum laden und Speichern. Beim Laden einer Karte kann man die Karte in einem Dialog auswählen. Zu beachten ist, daß nur Dateien mit der Endung ".map" angezeigt werden.

Beim Anwählen des Eintrags "Speichern" im Dateimenü wird die Karte unter dem zuletzt genutzten Dateinamen (der Name unter dem die Karte geladen bzw. zuletzt gespeichert wurde) abgespeichert. Wenn dieser Name nicht existiert, passiert dasselbe wie bei "Speichern unter". Dabei wird auch ein Dialog geöffnet, in dem man das Verzeichnis und den Dateinamen zum Speichern auswählt. Zu beachten ist hier ebenfalls, daß automatisch eine Endung ".map" angehängt wird, falls man sie nicht explizit ausschreibt.

3. Editieren der Karte

3.1. Plazieren von Terrain und Objekten

Nach dem Laden oder Erzeugen einer Karte will man vermutlich als Erstes das Terrain verändern und einige Gebäude oder Armeen plazieren. dazu gibt es verschiedene Schaltflächen auf der rechten Seite des Editors.

Mit den Terrainschaltern wählt man aus, welches Terrain man setzen möchte. Daneben gibt es noch verschiedene andere Schaltflächen für:

- Setzen des Terrains in einem 1x1-Gebiet
- Setzen des Terrains in einem 3x3-Gebiet
- Entfernen von Objekten (das rote Kreuz); Zu beachten ist, daß man nicht gefragt wird, ob man etwas sicher löschen möchte, ferner sollte man beachten, daß Armeen immer zuerst entfernt werden!
- Plazieren von Armeen, Städten, Ruinen und Tempeln

Bei den meisten Objekten wird eine elementare Fehlerprüfung durchgeführt. So ist es z.B. nicht möglich, Städte im Wasser zu plazieren, das Terrain wird automatisch zu Gras umgeändert. Dies klappt allerdings nicht für Armeen, also sollte man sicherstellen, daß Armeen nur dann im Wasser stehen, wenn sie aus Einheiten bestehen, die sich im Wasser bewegen können.

3.2. Ändern von Objekteigenschaften

Im Allgemeinen kann man die Eigenschaften eines Gebäudes/einer Armee durch Rechtsklick darauf ändern. Wenn mehrere Objekte sich übereinander befinden (z.B. eine Armee auf einer Ruine), so wird man gefragt, was man editieren möchte.

3.2.1. Städte

Ein Rechtsklick auf eine Stadt öffnet eine Dialog zum Editieren derselben. Das Meiste davon sollte selbsterklärend sein, deswegen werde ich nur einige Besonderheiten beschreiben.

Der Unterschied zwischen einfachen und speziellen Produktionen ist bereits in der Spielanleitung beschrieben worden (einfach = standard Armeeset; speziell = Armeeset des Spielers, wird entfernt wenn Stadt erobert wird).

Eine weitere Bemerkung soll der Asuwahlbox gelten, in der man zwischen "unberührt" und "niedergebrannt" auswählen kann. Wenn ein Spieler eine Stadt erobert, kann er sie zerstören. Manchmal möchte man eine Stadt schon von Beginn an zerstört haben. Um das zu erreichen, wählt man hier "niedergebrannt" aus.

3.2.2. Ruinen und Tempel

Was man hier ändern kann, ist hauptsächlich der Name. Im Fall von Ruinen kann man außerdem den Bewohner (den der Held beim Durchsuchen bekämpfen muß) auswählen, indem man auf die Schaltfläche "Bewohner" klickt. Der folgende Dialog ist identisch mit dem zum Editieren einer Armee.

3.2.3 Armeen

Ein Rechtsklick auf eine Armee öffnet den entsprechenden Dialog. Dieser sollte wieder zum größten Teil selbsterklärend sein.

Wenn man den Eigentümer einer Armee ändert, sollte man sicherstellen, daß alles konsistent ist. Zum Beispiel sollte die Armee eines Spielers nicht in einer neutralen Stadt herumstehen. Schließlich ist der Zweck des Editors `_nicht_`, die Fehlerbehandlung des Spiels herauszufordern. 😊

Die Eigenschaften der einzelnen Einheiten können ebenfalls geändert werden. Die Idee dahinter ist, daß manchmal ein wenig Editieren notwendig ist, um z.B. verwundete oder erfahrene Einheiten darzustellen.

4. Einige andere Menüeinträge

Die anderen Menüeinträge, die nirgendwo anders beschrieben werden, sind:

- Ein Dialog zum Finden von Objekten; Man gibt die ID des Objekts, das man sucht, ein (das Objekt kann eine Stadt, Ruine, ein Tempel, eine Armee oder eine Einheit sein), drückt "Enter" und die Karte springt an die entsprechende Stelle.
- Reports für Städte und Armeen. Durch Klicken auf eines der dargestellten Objekte gelangt man an dessen Position

Die anderen Einträge (Spieler und Ereignisverwaltung) sind Gegenstand der folgenden Kapitel.

5. Verwalten von Spielern

Durch Auswahl des Eintrags "Spieler" vom Menü "Edit" öffnet man einen Dialog zum Verwalten der Spieler.

Durch Klicken auf "Nächster" bzw. "Voriger" kann man alle Spieler durchgehen, mit Hilfe der 2 Dropdowns den Spielertyp und das Armeeset festlegen. Es gibt 4 Spielertypen: Mensch, Einfache KI (macht gar nichts), Schnelle KI (erstellt Armeen und greift an) und clevere KI (verteidigt auch Städte etc.).

Die Eingabefelder darunter sollten selbsterklärend sein. Man kann die Spielerfarbe, den Namen und den anfänglichen Goldvorrat festlegen.

Es gibt immer einen neutralen Spieler, den man nicht entfernen kann (und sollte). Weiterhin kann man Spieler unsterblich machen ("persistent", d.h. sie sterben nicht, wenn sie keine Städte mehr haben). Bei der schnellen KI kann man speziell noch einstellen, ob sie Armeen vereint, und man kann sie auf wahnsinnig (maniac) setzen. Im letzteren Fall greift sie relativ unterschiedslos alles in ihrer Nähe an (nett für wandernde Monster).

Wenn man einen neuen Spieler hinzufügt, wird man gefragt, ob man ihn vor den aktuellen Spieler oder ans Ende der Liste einfügen will, damit man etwas mehr Kontrolle hat.

6. Verwalten von Ereignissen

6.1. Einführung in das Ereignissystem

Wenn man ein Szenario erzeugt, möchte man üblicherweise spezielle Sachen machen. So sollte es zum Beispiel Armeen im Hinterhalt geben oder ungewöhnliche Siegbedingungen (z.B. Durchsuchen einer speziellen Ruine oder das Befördern einer speziellen Einheit an eine spezielle Position). In FreeLords wird das mit dem integrierten Ereignissystem gemacht. Es ist sehr mächtig, aber auch relativ kompliziert und benötigt eventuell so etwas wie Programmierfähigkeiten (zumindest im Denken).

Bevor ich in die Details gehe, sei hier eine kurze Erklärung gegeben. Das Ereignissystem besteht aus drei unterschiedlichen Teilen: Ereignisse, Reaktionen und Bedingungen.

- Ereignisse werden durch spezielle, nun ja, Ereignisse ausgelöst. Beispielsweise wird ein Ruinen-Ereignis ausgelöst, wenn ein Spieler eine bestimmte Ruine durchsucht.
- Jedem Ereignis können eine oder mehrere Reaktionen zugeordnet werden. Wenn das Ereignis ausgelöst wird, passiert das gleiche mit allen Reaktionen. So eine Reaktion kann z.B. "Ein bestimmter Spieler stirbt" oder "Das Spiel wird gewonnen" sein.
- Jedes Ereignis und jede Reaktion kann noch eine oder mehrere Bedingungen haben. Das Ereignis oder die Reaktion wird nur ausgelöst, wenn eine bestimmte Bedingung wahr ist. Um z.B. sicherzustellen, daß ein Eroberungs-Ereignis nur dann ausgelöst wird, wenn ein bestimmter Spieler die Stadt erobert, kann man eine Bedingung nutzen, die prüft, ob dieser Spieler gerade am Zug ist.

Jetzt ein paar mehr Details:

Ereignisse haben eine eindeutige ID (man muß sie manchmal identifizieren), eine Liste von Bedingungen und Reaktionen, einige (typspezifische) Daten und einen Kommentar. Der Kommentar hat keine Relevanz für das Spiel, aber man kann damit das Ereignis erklären, so daß andere Leute verstehen, wofür das Ereignis da ist. Daher sollte er am besten in Englisch sein. Weiterhin kann ein Ereignis am Anfang des Spiels im aktivierten oder deaktivierten Zustand sein. Wenn ein Ereignis ausgelöst wird (was nur passiert, wenn das Ereignis aktiviert ist), wird das Ereignis deaktiviert. Ereignisse werden das ganze Spiel hindurch nie gelöscht, so daß es möglich ist, ein Ereignis später wieder zu aktivieren.

Bedingungen haben lediglich Daten (z.B. die ID des Spielers, der am Zug sein muß). Wenn das Ereignis oder die Reaktion, zu der die Bedingung gehört, ausgelöst wird, prüft es zuerst, ob die Bedingung wahr zurückliefert; wenn nicht, wird das Ereignis/die Reaktion nicht ausgelöst. Im Fall eines Ereignisses, bleibt dieses anschließend aktiviert, wenn die Bedingung nicht erfüllt ist.

Reaktionen können schließlich Bedingungen und verschiedene Daten haben. Wichtig zu wissen ist, daß die Reaktionen eines Ereignisses in der Reihenfolge, in der sie auftreten, ausgelöst werden. Das ist bedeutsam, wenn man z.B. eine Armee plazieren, ein paar Mitteilungen zeigen und die Armee wieder entfernen möchte.

Viele Ereignisse, Reaktionen und Bedingungen benötigen die ID eines Objektes als Parameter. Wenn ein Objekt mit dieser ID nicht existiert, werden sie normalerweise einfach nie ausgelöst. Im Fall von Bedingungen kann das allerdings Probleme bereiten; einige liefern dann grundsätzlich wahr, andere falsch zurück.

6.2. Editieren von Ereignissen

Es gibt drei Dialoge zum Editieren von Ereignissen, Reaktionen und Bedingungen. Jeder von ihnen hat ein paar spezielle Eigenschaften, die ich hier kommentieren möchte.

Grundsätzlich gilt, daß wenn man ein Ereignis oder eine Reaktion hinzufügt, eine Liste von Wahlmöglichkeiten erscheint, aus der man den Typ des Ereignisses/der Reaktion auswählt. Dieser Typ kann später nicht mehr geändert werden! Man muß das Ereignis/die Reaktion erst entfernen und anschließend ein(e) Neue(s) hinzufügen.

Der Dialog zum Editieren der Ereignisse zeigt erst einmal alle bereits hinzugefügten Ereignisse an. Man kann eins davon auswählen, seine Daten editieren etc. Durch Klicken auf die einzelnen Schaltflächen kann man Reaktionen und Bedingungen hinzufügen/entfernen sowie neue Ereignisse hinzufügen bzw. vorhandene entfernen. Neue Ereignisse werden immer an das Ende der Liste angehängt.

Durch Klicken auf die "Reaktionen"-Schaltfläche kann man den Dialog zum Editieren derselben hervorzaubern. Dieser ist zum größten Teil mit dem Ereignisdialog identisch, es gibt also wieder Eingabefelder zum Editieren der Daten, man kann Bedingungen editieren etc. Da jedoch die Reihenfolge hier wichtig ist (die Reaktionen weiter oben in der Liste werden auch zuerst ausgelöst), ist das Hinzufügen neuer Reaktionen ein bißchen anders. Möchte man eine Reaktion hinzufügen, so wählt man _zuerst_ die Reaktion aus der Liste aus, _vor_ der die neue Reaktion eingefügt werden soll und klickt anschließend auf "Hinzufügen". Um eine Reaktion an das Ende der Liste anzuhängen, klickt man auf den letzten Eintrag in der Liste und anschließend ebenfalls auf "Hinzufügen".

Der Dialog für die Bedingungen hat eine weitere Eigenart. Wenn man eine Bedingung hinzufügt, wird eine Standardbedingung an das Ende der Liste angehängt. Der Typ kann mit Hilfe einer Auswahlliste jederzeit geändert werden.

6.3. Liste aller Ereignisse

Zur Zeit gibt es 11 unterschiedliche Arten von Ereignissen. Da eine Übersetzung hier ziemlich dämlich wäre, habe ich die englischen Bezeichnungen benutzt. Ich hoffe, daß jeder, der den Editor nutzt, hinreichende Englischkenntnisse hat, um sich die Bedeutung auch ohne Nachschlagewerk zusammenzureimen.

KillAll	Dieses Ereignis wird ausgelöst, wenn alle Spieler bis auf einen gestorben sind. Der neutrale Spieler zählt hierbei nicht mit.
PlayerDead	Dieses Ereignis wird ausgelöst, wenn ein bestimmter Spieler stirbt. Als Parameter bekommt es die ID des Spielers.
CityConq	Dieses Ereignis wird ausgelöst, wenn eine Stadt erobert oder zerstört wird und bekommt die ID der betreffenden Stadt als Parameter. Zu beachten hier ist, daß eine zerstörte Stadt nicht wieder erobert werden kann, und Städte im Verlauf eines Spiels zerstört werden können. Man sollte also sicherstellen, daß das nicht passiert (oder das Ereignis nicht zu wichtig ist).
ArmyKilled	Wenn eine Einheit stirbt, wird dieses Ereignis ausgelöst. Es bekommt die ID der betreffenden Einheit als Parameter.
Round	Dieses Ereignis wird ausgelöst, wenn eine bestimmte Spielrunde beginnt. Es bekommt die Rundenummer als Parameter mit. Dieses Ereignis sollte in Verbindung mit der Player-Bedingung genutzt werden, um Reaktionen zu Beginn der Runde eines bestimmten Spielers auszulösen.
RuinSearch	Wird ausgelöst, wenn eine Ruine durchsucht wird. Bekommt die ID der Ruine als Parameter.

TempleSearch	Dasselbe, aber diesmal für Tempel.
Dummy	Dieses Ereignis wird nie ausgelöst. Es kann vor allem in Verbindung mit der RaiseEvent-Reaktion und der Counter-Bedingung recht nützlich werden. In Abschnitt 6.6. wird ein Beispiel dafür gegeben.
NextTurn	Dieses Ereignis ist dem Ereignis Round ziemlich ähnlich, aber wird bei der nächsten sich bietenden Gelegenheit ausgelöst. Zusammen mit der Player-Bedingung und der ActEvent-Reaktion kann man dieses Ereignis nutzen, um einem speziellen Spieler mitzuteilen, daß etwas passiert ist. Hier sei ebenfalls auf Abschnitt 6.6. verwiesen.
StackKilled	verhält sich wie ArmyKilled, aber hier wird auf den Tod einer Armee (engl. stack) gewartet. Dieses Ereignis ist redundant (man kann es z.B. mit dem ArmyKilled-Ereignis und etwas mehr simulieren), aber kann eventuell als Kurzfassung nützlich sein. Wenn man dieses Ereignis nutzt, sollte man sich <u>unbedingt</u> vor Augen halten, daß Armeen sich vereinigen und teilen können bzw. aufgefüllt werden können. Man sollte dieses Ereignis also nur nutzen, wenn man sich absolut sicher ist, daß das mit der betreffenden Armee nicht passiert. Dieses Ereignis bekommt die ID der Armee als Parameter.
StackMove	Dieses Ereignis wird immer dann ausgelöst, wenn eine Armee den als Parameter angegebenen Punkt überschreitet.

6.4. Liste aller Reaktionen

Zur Zeit existieren 15 Reaktionen (wieder englische Nomenklatur):

Message	Wahrscheinlich <u>die</u> Reaktion schlechthin. Es zeigt lediglich einen Text in einem Fenster an. Dieser Text ist auch der Parameter.
AddGold	Diese Reaktion fügt dem Goldvorrat eines Spielers etwas Gold hinzu. Sie hat zwei Parameter, die ID des betroffenen Spielers und die Menge an Gold. Trotz des Namens kann man dem Spieler auch Gold wegnehmen, indem man einen negativen Betrag eingibt.
AddUnit	Diese Reaktion fügt dem Spiel eine Armee hinzu. Die Parameter sind u.a. die ID des Spielers, dem die Armee anschließend gehört und die Position, wo die Armee auftaucht. Schließlich kann die Armee durch Klicken auf die Schaltfläche "Armee" zusammengestellt werden. Wenn die angegebene Position blockiert ist, versucht diese Reaktion, die Armee auf einem der unliegenden Felder abzusetzen. Ist dies nicht möglich, gibt sie auf und plaziert die Armee gar nicht (das ist ein Bug in FreeLords).
DelUnit	Beseitigt eine bestimmte Einheit. Die ID der betreffenden Einheit ist hier der Parameter.
Update	Diese Reaktion hat keinen Parameter. Sie aktualisiert lediglich die Spielfeldansicht. Wenn man z.B. eine Armee auf dem Spielfeld plaziert (per AddUnit) und anschließend eine Message-Reaktion nutzt, wird die Armee auf der Karte nicht angezeigt. Wenn man diese Reaktion zwischendurch nutzt, ist die Einheit anschließend auf der Karte zu sehen.
Center	Zentriert die Karte auf einer bestimmten Position und aktualisiert nebenbei die Karte.
CenterObj	Wie Center, aber statt der Position gibt man die ID eines Objektes an. Das Objekt kann eine Stadt, Ruine, Armee, Einheit oder ein Tempel sein. Nützlich, wenn man die Karte z.B. auf einer bestimmten Einheit zentrieren möchte.
CreateItem	Erzeugt an einer anzugebenden Stelle ein Artefakt. Die Parameter sind die Position und der Index des Artefakts. Da man Artefakte zur Zeit nicht anschauen kann, kann

ich nur auf die Datei items.xml verweisen, wo der Index zwischen den Tags `<d_index>` und `</d_index>` zu finden ist.

WinGame	Beendet das Spiel mit einem Sieg. Als Parameter kann man einen Status mitgeben. Der wird zur Zeit nicht benutzt, aber kann später in nichtlinearen Kampagnen ganz nützlich sein. Zur Zeit kann man ihn getrost ignorieren.
LoseGame	Dasselbe wie WinGame. Der einzige Unterschied ist, daß kein Siegbildschirm, sondern eine Niederlage am Ende angezeigt wird. Da diese Bilder aber zur Zeit noch nicht existieren, ist das noch egal. Das sollte sich aber bald ändern. 😊
RaiseEvent	Mit dieser Reaktion kann man ein anderes Ereignis auslösen. In Abschnitt 6.6. gibt es ein Beispiel dafür. Als Parameter muß man die ID des betreffenden Elements angeben.
ActEvent	Aktiviert/Deaktiviert ein Ereignis. Diese Reaktion hat zwei Parameter, die ID des Ereignisses und ob das Ereignis aktiviert (1) oder deaktiviert (0) werden soll. Ein Beispiel für diese Reaktion sind Mehrfachereignisse. Normalerweise wird ein Ereignis beim Auslösen deaktiviert. Wenn sich aber unter den Reaktionen eine befindet, die das Ereignis wieder aktiviert, kann dieses unendlich oft ausgelöst werden.
RevivePlayer	Bringt einen toten Spieler wieder unter die Lebenden. Als Argument bekommt diese Reaktion die ID des Spielers. Achtung: Der wiedererweckte Spieler sollte unbedingt eine Stadt bekommen (siehe TransferCity weiter unten), ansonsten wird er ziemlich bald wieder sterben.
KillPlayer	Der Name ist Programm. Als Parameter ist die ID des zu tötenden Spielers anzugeben. Alle Armeen des Spielers werden dabei entfernt und alle Städte dem neutralen Spieler zugeschlagen.
TransferCity	Läßt eine Stadt den Besitzer wechseln. Notwendige Parameter sind offensichtlich die ID des neuen Besitzers und der Stadt selber.

6.5. Liste aller Bedingungen

Es gibt zur Zeit ebenfalls 5 Bedingungen, nämlich:

Player	Prüft, ob ein bestimmter Spieler an der Reihe ist. Z.B. in Verbindung mit dem Round-Ereignis nützlich, um den Zug eines bestimmten Spielers abzuwarten. Der Parameter ist die Spieler ID.
Counter	Diese Bedingung ist etwas eigenes. Als Parameter hat es einen Zähler, der immer, wenn die Bedingung abgefragt wird, um eins dekrementiert wird. Wenn er schließlich 0 erreicht, liefert diese Bedingung immer wahr zurück.
Living	Prüft, ob ein bestimmter Spieler noch lebt. Parameter ist die ID des Spielers.
Dead	Prüft, ob ein bestimmter Spieler tot ist. Parameter ist wieder die Spieler ID.
Army	Diese Bedingung bekommt die ID einer Einheit und prüft, ob die Einheit in der gerade aktiven Armee ist (das ist die Armee, die gerade etwas macht, z.B. sich bewegen, eine Ruine durchsuchen etc.). Zum Beispiel in Verbindung mit dem StackMove Ereignis nützlich, um zu prüfen, ob eine bestimmte Einheit an eine bestimmte Stelle bewegt worden ist.

6.6. Einige Beispiele

a) Standardsetup

- Als erstes Beispiel möchte ich das Standardsetup eines Zufallsspieles beschreiben. Ich nehme hier an, daß es sowohl menschliche als auch Computerspieler gibt.

Gewollt ist folgendes: Wenn ein Spieler stirbt, sollte jeder andere (menschliche) Spieler über dessen Tod informiert werden. Wenn ein menschlicher Spieler alle anderen besiegt hat, so sollte das Spiel mit einem Sieg, sonst mit einer Niederlage enden.

Die Realisierung geht etwa so. Für den Tod eines jeden Spielers X erzeugen wir ein (deaktiviertes) NextTurn-Ereignis für jeden menschlichen Spieler (außer X, falls X menschlich ist) mit einer Message-Reaktion, die sagt "Spieler X ist tot". Anschließend erzeugen wir ein PlayerDead-Ereignis mit X' ID und Reaktionen, die jedes dieser NextTurn-Ereignisse aktivieren. Auf diese Weise werden alle menschlichen Spieler informiert, wenn X gestorben ist.

Schließlich wird noch ein KillAll-Ereignis erzeugt. Dieses Ereignis hat, für jeden Spieler X außer dem neutralen Spieler, zwei Reaktionen, eine Message- und eine WinGame- (falls X menschlich ist) oder LoseGame-Reaktion (falls X eine KI ist). Beide Reaktionen haben jeweils eine Living-Bedingung mit der ID von X.

Achtung: Wenn man keinerlei Ereignisse für ein Szenario erzeugt, wird das Spiel nie enden! Daher sind einige Ereignisse zum Gewinnen/Verlieren unabdingbar!

b) Benutzung des Dummy-Ereignisses und der Counter-Bedingung

- Hier möchte ich eine Situation beschreiben, wo die Counter-Bedingung sehr praktisch ist. Man stelle sich eine Karte mit drei Ruinen vor. Wir möchten, daß der Spieler das Szenario gewinnt, wenn er alle drei Ruinen durchsucht hat. Wie macht man das?

Die Antwort ist einfach. Man erzeugt ein Dummy-Ereignis mit einer WinGame-Reaktion und einer Counter-Bedingung. Den Zähler setzt man hierbei auf drei. Jetzt erzeugt man für jede Ruine ein RuinSearch-Ereignis (mit der Bedingung, daß der Spieler am Zug ist) mit einer RaiseEvent-Reaktion. Diese Reaktion bekommt die ID des Dummy-Ereignisses.

Was passiert nun? Wann immer der Spieler eine der Ruinen durchsucht, wird das entsprechende RuinSearch-Ereignis ausgelöst. Dessen Reaktion löst dann das Dummy-Ereignis aus. Da eine Ruine nur einmal durchsucht werden kann, wird der Zähler der Counter-Bedingung für jede durchsuchte Ruine einmal dekrementiert. Wenn alle drei Ruinen durchsucht wurden, erreicht der Zähler schließlich 0 und das Spiel wird gewonnen.

Zu beachten hierbei ist, daß in diesem einfachen Beispiel der Fall, daß ein anderer Spieler die Ruine durchsucht, nicht behandelt wird. In diesem Fall würde das Spiel nie enden, so daß man das normalerweise auch abfangen sollte.

7. Manuelles Editieren des Szenarios

Wenn man einige exzentrische Sachen machen möchte, die im Editor zur Zeit noch nicht implementiert sind, gibt es zwei Möglichkeiten

Erstens kann man eine Mail, bevorzugt an die Entwicklerliste, schicken. Da vermutlich nur wenige Leute ernsthaft Karten für FreeLords basteln werden, sind wir wahrscheinlich so überglücklich darüber, daß wir das Feature relativ schnell einbauen werden. Dann muß man nur noch auf ein neues Release warten (oder die CVS-Version) nehmen und ist fertig.

Manchmal möchte man aber nicht warten oder z.B. muß man die Szenariodatei auf eine neue Version bringen.

Editoranleitung - Freelords Wiki

In dem Fall kann man das Szenario nur manuell editieren. Bitte nicht abschrecken lassen, es ist einfacher als es klingt!

Die Szenariodatei nutzt ein spezielles XML-Format. Einfach mit einem Texteditor öffnen und man wird wahrscheinlich große Teile gleich wiedererkennen. Zum Editieren sollte man drei Dinge beachten (abgesehen von einer Sicherungskopie!!!):

1. Jedes Objekt im Spiel (Armeen, Städte etc.) speichert seine ganzen Daten zwischen zwei Tags. Zum Beispiel speichert ein Spieler seine gesamten Daten zwischen den Tags <player> und </player>. Innerhalb dieser Tags werden auch alle möglichen Unterobjekte gespeichert. Eine Armee (stack), die aus Einheiten (army) besteht, sieht dann etwa so aus:

```
<stack>

  ♦ ..Armeedaten...

    <army>

      ♦ ..Daten der ersten Einheit...
    </army>

    <army>

      ♦ ..Daten der zweiten Einheit...
    </army>
  ♦ ..
</stack>
```

Die tatsächlichen Daten werden immer zwischen Tags, die mit "d_" starten, gespeichert. Z.B. sieht ein Player-Tag etwa so aus:

```
<player>

  ♦ <d_id>4</d_id>

  <d_name>Fürst der Finsternis</d_name>
  ♦ ..
</player>
```

Das ist der größte Teil, den man wissen muß.

2. Die Bedeutung sämtlicher Tags wird in der Datei Savefile im Dokumentationsverzeichnis erklärt. Man sollte gegen Ende mit dem Lesen anfangen, der Text darüber beschreibt jeweils die Änderung. Die Datei existiert übrigens auch nur auf englisch.
3. Falls sich die Version ändert... Die Versionsnummer wird im ersten Tag abgespeichert. Durch Ändern auf einen neuen Wert kann man den Parser dazu bringen, die Karte zu akzeptieren. Trotzdem sollte man sicherstellen, daß man dabei die Änderungen, die in der Datei Savefile für die neue Versionsnummer angegeben sind, ebenfalls manuell einpflegt, ansonsten können grausame Fehler auftreten. Wenn du die Karte zu unserem Spiel beitragen möchtest und ein derartiges Problem hast, würden wir auch die Änderung selber machen.

Editoranleitung - Freelords Wiki

Das ist alles. Wenn du es ein bißchen unbefriedigend findest, daß das Dokument hier schon aufhört, solltest du dir klarmachen, daß Open Source von Mithilfe lebt, und das Minimum ist eine Rückmeldung, was unklar ist oder fehlt. 😊

Editoranleitung (last edited 2007-04-16 11:19:15 by ulf)

