I L L A R I O N -Dokumentation MAP-System-

(nur für internen Gebrauch bestimmt)
Die im Folgenden aufgeführten Informationen sind noch unvollständig und sollen ergänzt werden. Für Hinweise auf Fehler, Anregungen und Kritik danke ich im Voraus.

Illarion_initmap

Zentrale Datei ist zunächst einmal die illarion_initmaps. Aus ihr erhält der Server beim Start den Hinweis darauf, welche Maps vorhanden sind und geladen werden müssen. Beim ordnungsgemäßen Anhalten des Servers werden diese Einträge überschrieben. Das heißt andererseits, Änderungen an der Datei können nur bei Server-Down durchgeführt werden, da sonst beim nächsten Server-Stop mit den alten Daten aus dem Arbeitsspeicher überschrieben wird.

Im Folgenden wird der Aufbau der Datei illarion_initmap beschrieben, wobei der Inhalt natürlich inzwischen anders aussehen mag. Alle Wertangaben beziehen sich somit nur auf diese zugrundegelegte Version vom Januar 2004.

Datei-Inhalt nach Hex-Dump:

```
29 00
00 00 E7 FF E9 FF 40 01 03 01 01 01 00 6D 00 80
00 0D 00 0C 00 01 01 00 5B 00 56 00 0A 00 09 00
01 FD FF 60 00 63 00 07 00 14 00 01 01 00 5F 00
63 00 12 00 14 00 01 02 00 60 00 63 00 11 00 08
00 01 01 00 4A 00 6B 00 0F 00 0C 00 01 01 00 54
00 7F 00 09 00 07 00 01 02 00 54 00 7F 00 09 00
07 00 01 01 00 79 00 5D 00 17 00 17 00 01 FD FF
F0 FF 02 00 14 00 14 00 01 FD FF 0F 00 05 00 19
00 19 00 01 FE FF 14 00 09 00 08 00 08 00 01 FD
FF F6 FF 37 00 19 00 19 00 01 FD FF 2E 00 63 00
32 00 32 00 01 FD FF EC FF B4 00 29 00 29 00 01
01 00 37 00 71 00 0F 00 0C 00 01 02 00 37 00 71
00 0E 00 0C 00 01 01 00 FB FF C3 00 0C 00 0C 00
01 02 00 FB FF C3 00 0C 00 0C 00 01 FD FF F0 00
00 00 28 00 28 00 01 FA FF 78 00 00 00 96 00 96
00 01 01 00 E0 00 09 00 10 00 0E 00 01 01 00 21
00 94 00 0F 00 0E 00 01 01 00 8F 00 C7 00 17 00
1B 00 01 FF FF 8F 00 C7 00 17 00 1B 00 01 01 00
0A 00 A1 00 09 00 0B 00 01 01 00 E7 00 25 00 03
00 03 00 01 01 00 E9 00 1E 00 03 00 03 00 01 01
00 DE 00 1B 00 03 00 03 00 01 02 00 E2 00 0B 00
0C 00 0A 00 01 01 00 F2 FF 28 00 0A 00 0D 00 01
```

```
01 00 F8 FF A0 00 0C 00 0B 00 01 01 00 F7 FF AF
00 08 00 07 00 01 01 00 EB FF A4 00 0A 00 0C 00
01 01 00 ED FF 98 00 0A 00 08 00 01 01 00 F8 FF
96 00 08 00 08 00 01 01 00 0E 00 60 00 16 00 14
00 01 01 00 FB FF 53 00 10 00 11 00 01 FD FF FB
FF 54 00 10 00 10 00 01 FD FF 0F 00 61 00 14 00
12 00 01
```

Erläuterung/Aufbau:

29 Angabe über die Anzahl der einzulesenden Datensätze (= Anzahl vorhandener Maps)
h29 = 41 Datensätze

Eine Map besteht aus jeweils drei Dateien mit der Namensgebung
Illarion_ [EBENE]_ [X-Koordinate]_ [Y-Koordinate]_[Zusatz]
mit den Varianten bei Zusatz: _map, _item und _container.

Wir beschäftigen uns jetzt nur mit der *_map -Variante, die die Angaben zur Lage und Größe der Map und den darin befindlichen Bodentiles enthält.

Beispiel: Illarion_ 2_ 96_ 99_map

Diese Map hat also ihren Ausgangspunkt in X=96, Y=99, Z=(+)2

In der Hex-Übertragung wären dies die Bytefolgen: 02 / 60 / 63

wir finden 02 00 60 00 63 00
Die Schreibweise folgt also der Intention Low-Byte / High-Byte.
Die Reihenfolge folgt der Konvention des Dateinamens unter Erstnennung der Ebene.

Wir finden in der Datei, nach Abzug eines Byte-Paares für die Datensatz-Anzahl, weitere 451 Byte.

Vom Führungsbyte wissen wir, dass es sich nur um 41 Datensätze handelt, also müssen zu einem Datensatz weitere Angaben gehören.

In den Bytes davor finden wir die Kombination 01 00 5F 00 63 00, was der Map mit der Kennung Illarion_ 1_ 95_ 99_map entspricht. Zwischen diesen beiden Abschnitten befinden sich die Bytes 12 00 14 00 01, also dezimal 18 / 20 / 1 .

Der GM-Befehl zur Einrichtung einer Map kennt neben den Angaben des Ursprung und der Ebene auch die Angaben zur X/Y-Ausdehnung. Die beiden Paare 12 00 und 14 00 bilden diese Angabe ab.

Damit besteht unsere Dateiangaben nunmehr aus wenigstens 5 Byte-Paaren (= 410 Byte)

Bei der Annahme von 11 Byte pro Datensatz kommen wir auf 41 x 11 = 451 Byte.
Das einzelne Byte 01 ist also entweder als Startbyte des Datensatzes zu werden, oder aber als letztes Byte.

Sortieren wir den Hexdump nach diesem Schema um, ergibt sich eine lesefreundliche Darstellung der einzelnen Datensätze:

```
29 00
00 00 E7 FF E9 FF 40 01 03 01 01
01 00 6D 00 80 00 0D 00 0C 00 01
01 00 5B 00 56 00 0A 00 09 00 01
FD FF 60 00 63 00 07 00 14 00 01
01 00 5F 00 63 00 12 00 14 00 01
02 00 60 00 63 00 11 00 08 00 01
01 00 4A 00 6B 00 0F 00 0C 00 01
01 00 54 00 7F 00 09 00 07 00 01
02 00 54 00 7F 00 09 00 07 00 01
01 00 79 00 5D 00 17 00 17 00 01
FD FF F0 FF 02 00 14 00 14 00 01
FD FF 0F 00 05 00 19 00 19 00 01
FE FF 14 00 09 00 08 00 08 00 01
FD FF F6 FF 37 00 19 00 19 00 01
FD FF 2E 00 63 00 32 00 32 00 01
FD FF EC FF B4 00 29 00 29 00 01
01 00 37 00 71 00 0F 00 0C 00 01
02 00 37 00 71 00 0E 00 0C 00 01
01 00 FB FF C3 00 0C 00 0C 00 01
02 00 FB FF C3 00 0C 00 0C 00 01
FD FF F0 00 00 00 28 00 28 00 01
FA FF 78 00 00 00 96 00 96 00 01
01 00 E0 00 09 00 10 00 0E 00 01
01 00 21 00 94 00 0F 00 0E 00 01
01 00 8F 00 C7 00 17 00 1B 00 01
FF FF 8F 00 C7 00 17 00 1B 00 01
01 00 0A 00 A1 00 09 00 0B 00 01
01 00 E7 00 25 00 03 00 03 00 01
01 00 E9 00 1E 00 03 00 03 00 01
01 00 DE 00 1B 00 03 00 03 00 01
02 00 E2 00 0B 00 0C 00 0A 00 01
01 00 F2 FF 28 00 0A 00 0D 00 01
01 00 F8 FF A0 00 0C 00 0B 00 01
01 00 F7 FF AF 00 08 00 07 00 01
01 00 EB FF A4 00 0A 00 0C 00 01
01 00 ED FF 98 00 0A 00 08 00 01
01 00 F8 FF 96 00 08 00 08 00 01
01 00 0E 00 60 00 16 00 14 00 01
01 00 FB FF 53 00 10 00 11 00 01
FD FF FB FF 54 00 10 00 10 00 01
FD FF 0F 00 61 00 14 00 12 00 01
```

Noch ein Wort zur Darstellung negativer Werte:

Negative Werte werden dargestellt, indem das Highbyte den Wert FF annimmt und im Low-Byte von FF=-1 ausgehend „zurückgezählt“ wird. FE FF ergibt somit den Dezimalwert -2. Daraus ergeben sich Konsequenzen für die in Illarion möglichen Koordinaten und Abmessungen einer Map.

Aufbau einer *_map-Datei

Als Nächstes sehen wir uns die zuvor mehrfach erwähnte Map-Datei an, in diesem Fall ist es die Illarion_ 1_ 95_ 99_map, mit 18 (h12) x 20 (h14) Feldern.

Hier zunächst wieder der Hex-Dump:

```
12 00 14 00 5F 00 63 00 01 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 10 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

28 00 30 70 28 00 30 70 28 00 30 40 28 00 30 70 28 00 20 40
28 00 20 70 28 00 20 70 29 00 20 40 29 00 20 40 29 00 20 40
29 00 20 40 29 00 20 40 29 00 30 40 29 00 30 40 29 00 30 40
29 00 30 40 29 00 30 40 29 00 30 40 29 00 30 40 00 00 00 00

28 00 30 70 28 00 30 00 28 00 30 00 28 00 30 00 28 00 30 00
28 00 20 00 28 00 20 70 29 00 20 00 29 00 20 00 29 00 20 00
29 00 20 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 30 70 2C 00 30 01 2D 00 30 00 30 00 30 00 33 00 30 00
28 00 30 00 28 00 20 40 29 00 20 00 29 00 20 00 29 00 20 00
29 00 20 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 40 FE FF 00 00

28 00 20 70 28 00 20 00 2E 00 20 00 31 00 20 00 34 00 30 00
28 00 30 00 28 00 30 70 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 40 FE FF 00 00

28 00 20 70 28 00 20 01 2E 00 20 00 31 00 20 00 34 00 30 00
28 00 30 00 28 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 40 FE FF 00 00

28 00 20 70 28 00 20 00 2E 00 20 00 31 00 20 00 34 00 30 00
28 00 30 00 28 00 30 70 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 40 FE FF 00 00

28 00 20 40 28 00 20 40 2E 00 20 00 31 00 20 00 34 00 30 00
28 00 30 00 28 00 30 70 29 00 30 40 29 00 30 40 29 00 30 40
29 00 30 40 29 00 30 40 29 00 30 00 29 00 30 40 29 00 30 40
29 00 30 40 29 00 30 40 29 00 30 40 29 00 30 40 FE FF 00 00

28 00 20 70 28 00 20 00 2F 00 20 00 32 00 20 00 35 00 30 00
```

```
28 00 30 00 28 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 20 70 28 00 20 70 28 00 20 70 28 00 20 00 28 00 30 70
28 00 30 70 28 00 30 70 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 20 70 28 00 20 00 28 00 20 00 28 00 20 00 28 00 30 00
28 00 30 00 28 00 30 70 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 40 29 00 30 40 29 00 30 40 29 00 30 40 29 00 30 40
29 00 30 40 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 20 70 28 00 20 00 28 00 20 00 28 00 20 00 28 00 30 00
28 00 30 00 28 00 30 70 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 40 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 20 40 28 00 20 00 28 00 20 00 28 00 20 00 28 00 30 00
28 00 30 00 28 00 30 70 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 40 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 20 70 28 00 20 00 2D 00 20 00 30 00 20 00 33 00 30 00
28 00 30 00 28 00 30 70 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 40 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 20 40 28 00 20 00 2E 00 20 00 31 00 20 40 34 00 30 00
28 00 30 00 28 00 30 70 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 40 29 00 30 40 29 00 30 40 29 00 30 40 29 00 30 40
29 00 30 40 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 20 70 28 00 20 00 2F 00 20 00 32 00 20 00 35 00 30 00
28 00 30 00 28 00 30 70 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 20 40 28 00 20 00 28 00 20 00 28 00 20 00 28 00 30 00
28 00 30 00 28 00 30 70 29 00 30 40 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 00
29 00 30 00 29 00 30 00 29 00 30 00 29 00 30 40 00 00 00 00

28 00 20 70 28 00 20 40 28 00 20 70 28 00 20 40 28 00 30 70
28 00 30 40 28 00 30 70 29 00 30 40 29 00 30 40 29 00 30 40
29 00 30 40 29 00 30 40 29 00 30 40 29 00 30 40 29 00 30 40
29 00 30 40 29 00 30 40 29 00 30 40 29 00 30 40 00 00 00 00
```

Der Header fällt sofort ins Auge: 12 00 14 00 5F 00 63 00 01 00

18 (h12) Felder in X-Richtung, 20 (h14) in Y-Richtung, Startkoorinate bei 95/98 (h5F/h63) und das Ganze auf Ebene 1.
Es handelt sich also um die Map der Zwischendecke des 1. OG der Bibliothek.

720 Byte-Paare folgen. 18 x 20 Felder = 360 Bytepaare. Ein Feld wird also von 2 Byte-Paaren beschrieben (4 Byte)

Wir wissen, dass der Fußboden innerhalb des Hauses aus Parkettboden (Id 40/ h28) besteht, auf der Dachterasse finden wir das graue Dachttile (Id 41/ h29) an.

Außerdem erkennen wir das Bodentile der Luke (Id 44/ h2C), dass von Parkett-Bodentiles benachbart ist.

Es fällt auf, dass die Tiles für Parkett und Dach nicht blockweise auftauchen, sondern wechselnd, was nichts anderes bedeuten kann, als dass die Auflistung nicht zeilenweise, sondern spaltenweise erfolgt. Durch das sich ergbende Muster wird jedoch unsere Vermutung, es handele sich um Doppelpaare pro Feld, nochmals bestätigt.

Die ersten zwanzig dieser Doppelpaare tragen den Wert 0. Es handelt sich also um den überstehenden Rand des „Daches“ der mit dem durchsichtigen Bodentile Id 0 belegt wurde. Es sind 20 Paare, entsprechend der Y-Ausdehnung. Ein weiterer Hinweis auf die spaltenweise Darstellung.

Bleibt noch die Frage nach der Bedeutung des zweiten Byte-Paares, bei dem auffällt, dass bestimmte Kombinationen dominieren.

Beim dritten Byte dominiert die h30/48, h20/32, h10/16 und 0, weitere Werte existieren nicht. Auffallend ist dabei die 16er-weise Regelmäßigkeit 0-16-32-48.

Beim vierten Byte scheinen neben der 0 vor allem die Werte h40/64 und h70/112.

Es gibt allerdings auch einige Ausreißer, die wir uns zunächst ansehen wollen.

Da wäre zunächst einmal eine Anordnung von Erst-Bytes mit den Werten **2D, 2E, 2F, 30, 31, 32, 33, 34 und 35,** die ins Auge fällt. 2D bis 35 also dezimal 45 bis 53 bezeichnen die Bodentiles für einen Teppich.

Regelmäßig erscheint noch das Auftauchen des Negativ-Wertes `FE FF`, also -2, was für die Bezeichnung eines Bodentiles eigentlich keinen Sinn ergibt. Der Wert taucht nur fünfmal auf, dafür aber in schönem gleichmäßigem Abstand von jeweils 20 Feldern in senkrechter Abfolge. Die so bezeichneten Tiles liegen auf der Map also waagerecht nebeneinander. Ein spaltenweise durchgeführter Umbruch zeigt, dass die Werte stehts am unteren Ende einer Spalte liegen, es handelt sich also um Bodentiles, die über den Rand des Gebäudes hinausragen. In dieser Darstellung wird nun auch sehr schön die Abgrenzung der Parketttiles von den südlicher liegenden Dachtiles deutlich.

Es handelt sich, aller Wahrscheinlichkeit nach, um die in der Projektbeschreibung dargestellten Specialfileds, die eingerichtet wurden, damit man durch Fenster und Türöffnungen hindurch in das Innere von Gebäuden blicken kann.
Tatsächlich liegen diese Felder alle im Eingangsbereich der Bibliothek.

Die Bedeutung des 3./4. Bytes sind noch nicht klar, ich vermute aber, dass es sich entweder um Erweiterungsraum handelt, der noch nicht genutzt wird, oder aber um die Beschreibung von Eigenschaften des jeweiligen Bodentiles. Wir werden sehen.

Aufbau einer *_item-Datei

Die zweite Map-Datei verrät ihren Inhalt schon in der Überschrift. Diese Datei enthält die auf der Map installierten Items, und zwar wohl überwiegend die unverrottbaren, fest installierten Teile wie Wände, Bäume, Hecken etc.

Werfen wir wieder einen Blick auf den Hex-Dump:

```
12 00 14 00 5F 00 63 00 01 00
```
(den Header brauchen wir nicht mehr zu erläutern)

```
Spalte 1
00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00

Spalte 2
01 F1 00 FF 01 | 01 21 00 FF 01 | 01 25 00 FF 01 | 01 21 00 FF 01
01 25 00 FF 01 | 01 21 00 FF 01 | 01 F0 00 FF 01 | 01 2D 01 FF 01
01 2D 01 FF 01 | 01 2D 01 FF 01 | 01 2D 01 FF 01 | 01 2D 01 FF 01
01 2D 01 FF 01 | 01 2D 01 FF 01 | 01 2D 01 FF 01 | 01 2D 01 FF 01
01 2D 01 FF 01 | 01 2D 01 FF 01 | 01 2D 01 FF 01 | 00

Spalte 3
01 1E 00 FF 01 | 00 00 00 00 00 | 01 20 00 FF 01 | 00 00 00 00 00
00 00 00 00 00 00 | 01 2D 01 FF 01 | 00

usw
01 1E 00 FF 01 00 00 00 00 00 01 26 00 FF 10 00 00 00 00 00
00 00 00 00 00 00 01 2D 01 FF 01 00

01 1E 00 FF 01 00 00 00 00 00 01 20 00 FF 01 00 00 00 00 00
00 00 00 00 00 00 01 2D 01 FF FA 00

01 1E 00 FF 01 01 3F 01 FF 01 00 00 00 00 01 56 00 FF 01 00
00 00 00 00 00 00 00 00 00 00 01 2D 01 FF 01 00

01 1E 00 FF 01 00 00 00 00 00 01 20 00 FF 01 00 00 01 D5 00
FF 01 01 D5 00 FF 01 01 D5 00 FF 01 00 01 D5 00 FF 01 01 D5
00 FF 01 01 D5 00 FF 01 00 00 01 2D 01 FF 01 00

01 3E 01 FF 01 01 41 01 FF 01 00 00 00 00 01 20 00 FF 01 02
2D 01 FF 01 27 01 FF 01 01 2D 01 FF 01 01 2D 01 FF 01 01 2D
01 FF 01 01 2D 01 FF 01 00 01 2D 01 FF 01 01 2D 01 FF 01 01
2D 01 FF 01 01 2D 01 FF 01 01 2D 01 FF 01 01 2D 01 FF 01 00

01 1E 00 FF 01 00 00 00 00 00 01 56 00 FF 01 00 00 00 00 00
```

```
00 00 00 00 00 00 01 2D 01 FF 01 00

01 EE 00 FF 01 02 1F 00 FF 01 10 01 FF 01 01 EF 00 FF 01 01
57 00 FF 01 02 EE 00 FF 01 10 01 FF 01 01 1F 00 FF 01 01 EF
00 FF 01 01 27 01 FF 01 00 00 01 D9 00 FF 01 01 D9 00 FF 01
01 D9 00 FF 01 01 D9 00 FF 01 01 D9 00 FF 01 01 D9 00 FF 01
00 00 01 2D 01 FF 01 00

01 1E 00 FF 01 00 00 00 00 00 01 20 00 FF 01 00 00 00 01 24
00 FF 01 01 24 00 FF 01 01 24 00 FF 01 01 24 00 FF 01 01 24
00 FF 01 01 24 00 FF 01 01 D8 00 FF 01 00 01 2D 01 FF 01 00

02 1E 00 FF 01 0A 01 FF 18 00 00 00 00 00 01 20 00 FF 01 00
00 00 00 00 00 00 00 01 24 00 FF 01 01 D8 00 FF 01 00 01 2D
01 FF 01 00

01 3E 01 FF 01 00 00 00 00 00 01 20 00 FF 01 00 00 00 00 00
00 00 00 01 24 00 FF 01 01 D8 00 FF 01 00 01 2D 01 FF 01 00

02 1E 00 FF 01 13 01 FF 01 00 00 00 00 00 01 20 00 FF 01 00
00 00 00 00 00 00 00 01 24 00 FF 01 01 D8 00 FF 01 00 01 2D
01 FF 01 00

01 3E 01 FF 01 01 D9 00 FF 01 01 D9 00 FF 01 01 24 00 FF 01
00 00 01 20 00 FF 01 00 00 00 01 24 00 FF 01 01 24 00 FF 01
01 24 00 FF 01 01 24 00 FF 01 01 24 00 FF 01 01 24 00 FF 01
01 D8 00 FF 01 00 01 2D 01 FF 01 00

02 1E 00 FF 01 0C 01 FF 01 00 00 01 D8 00 FF 01 00 00 01 20
00 FF 01 00 00 00 01 DB 00 FF 01 01 DB 00 FF 01 01 DB 00 FF
01 01 DB 00 FF 01 01 DB 00 FF 01 01 DB 00 FF 01 00 00 01 2D
01 FF 01 00

01 3E 01 FF 01                ; Feld belegt mit 1 Item 013E
00                            ; leeres Feld
00                            ; leeres Feld
01 D8 00 FF 01                ; Feld belegt mit 1 Item 00D8
00                            ; leeres Feld
00                            ; leeres Feld
01 20 00 FF 01                ; Feld belegt mit 1 Item 0020
02 11 01 FF 01 10 01 FF 01    ;Feld belegt mit 2 Items
                              013E/0110
00                            ; leeres Feld
00                            ; leeres Feld
00                            ; leeres Feld
00                            ; leeres Feld
00                            ; leeres Feld
00                            ; leeres Feld
00                            ; leeres Feld
00                            ; leeres Feld
00                            ; leeres Feld
00                            ; leeres Feld
```

```
01 2D 01 FF 01                ; Feld belegt mit 1 Item 012D
00                            ; leeres Feld

01 EE 00 FF 01 01 1E 01 FF 01
03 1F 00 FF 01 0D 01 FF 01 09 01 FF 01
01 1E 01 FF 01 01 1F 00 FF 01 01 1E 01 FF 01 01 EF 00 FF 01
01 2D 01 FF 01 01 2D 01 FF 01 01 2D 01 FF 01 01 2D 01 FF 01
01 2D 01 FF 01 01 2D 01 FF 01 01 2D 01 FF 01 01 2D 01 FF 01
01 2D 01 FF 01 01 2D 01 FF 01 01 2D 01 FF 01 01 2D 01 FF 01
00
```

976 Bytes für die Belegung von 360 Feldern (720 B.-Paare). Verbleiben also 256 Bytes.

Suchen wir also zunächst einmal wieder einige markante Eintragungen als Hilfspunkte.

Neben der Luke muss sich eine Leiter (item_id = 35 /h23) befinden, die auf das Dach (Ebene +2) führt. Die 23 dürfte demnach auch nur einmal auftauchen. Fehlanzeige!? Aber es ist ja auch keine Leiter, sondern eine Treppe (item_id = 229 /hE5). Wieder Fehlanzeige.

268, 269 sind die Ids für die beiden Kamine, also h 010C und h 010D. Wir finden ein `0D` umrahmt von zwei 01. Ist die Darstellung nun 01 0D oder 0D 01 ? Wir erinnern uns: In der illarion_initmap -Datei galt die Reihenfolge Lowbyte/Highbyte, aber gilt das hier auch?

In der Itemliste gibt es noch eine zweiten Eintrag „Treppe“ mit der ID 319 (h 01 3F). Diesmal werden wir fündig! `01 3F` gibt es tatsächlich, `3F 01` aber auch.

Schauen wir uns das erste Byte-Paar an, dass auf die 00-er Reihe folgt, die sich wohl daraus ergibt, dass es sich um die unbebauten durchsichtigen Felder des Bodentiles 0 handelt.
`01 F1 00. 01F1` -dezimal also 497 scheidet aus, wir haben derzeit nur 388 Item-Ids.
`F101` -dezimal 61697 macht noch viel weniger Sinn. Wo ist der Denkfehler?

Die einzige sinngebende Kombination wäre `00 F1`, also dezimal 241. Es würde dann wieder die Reihenfolge Lowbyte/Highbyte gelten und die ID müsste für ein Mauerteil stehen. Ein Blick in die Itemliste des Staff-Tools bestätigt die Vermutung.

Nach diesem Muster wollen wir nun die Stühle suchen, die sich auf der Dachterasse befinden. Stühle gibt es in vier Richtungsvarianten, die Ids sind: 212 bis 215 (hD4 bis hD7)
Wir finden eine Reihe von 6 Stühlen `D5 00`.

Depot (Id 321) `41 01` findet sich ebenfalls.

Häufig wiederkehrend zu erwarten ist auch die Hecke (Id 301/ h 12D). `2D 01`
Die erste Häufung weist 12 Hecken auf, die vermutlich die erste senkrechte Reihe bilden. Aus dem gleichbleibenden Abstand ist zu schließen, dass der Datensatz insgesamt 6 Byte umfasst, nämlich 2D 01 FF 01 01, wobei die letzte Hecke im letzten Byte abweicht: 2D 01 FF 01 0**0**

Zwischen unserem ersten Mauerteil `00 F1` und den Hecken `2D 01` liegen weitere Eintragungen, `21 00` , `25 00` und `F0 00` die es zu identifizieren gilt.
21 00 = Id 33 = Mauer
25 00 = Id 37 = Fenster
F0 00 = Id 240 = Mauer

Zusammen ergeben sich 19 Eintragungen plus ein Bodenfeld ID=0, was eine komplette Spalte darstellt.
Die zweite Spalte müsste ebenfalls mit einem Heckenstück enden. Für diese und die dritte Spalte ergeben sich aus der Abfolge eine Länge von jeweils 32 Byte.

18 Felder in X-Richtung ergeben 18 Spalten a 32 Byte = 576 Byte. Daraus folgert nichts anderes, als dass die Spalten unterschiedlich lang sein müssen, um insgesamt auf 976 Byte zu kommen.

wir markieren jetzt alle Hecken, da alle Spalten mit diesem Item enden müssten.
Da auf die letzte Hecke, die mutmaßlich süd-östliche Kante der Map drei Byte folgen, wird diese Reihenfolge nun für alle Spalten angenommen und ein entsprechender Umbruch durchgeführt.
Dadurch entstehen jetzt 17 Spalten, die bis auf eine Ausnahme alle auf die Endung FF 01 00 enden. Die äußerst unterschiedlichen Spaltenlängen werden hier nun sehr gut lesbar.
Die noch fehlende 18.Spalte rekrutiert sich aus der ersten Byte-Folge. Es handelt sich wieder um die unbebauten Felder des Boden-Tiles Nr. 0.

Aufgrund der Bebauung wissen wir, dass die Nordseite eine komplette Wand ergeben muss.

Nach dem Führungsbyte, dass stets die Zahl 1 oder 2 trägt, wiederholen sich die Eintragungen: neben dem schon als Mauer identifierten F1 00 finden sich:

```
1E 00 = Id 30  = Mauer
3E 01 = Id 318 = Fenster
EE 00 = Id 238 = Mauer
```

Die letzte Spalte muss eine Mauer entlang der Y-Koordinate aufweisen, die dann in eine Hecke übergeht und insgesamt über 20 Eintragungen verfügt. Zwölfmal finden wir die Hecke, wenn wir entsprechend umbrechen ergibt sich ein Schema, nachdem jedes Item 5 Byte beansprucht. Dem Item-Identifier geht ein Führungsbyte voraus.

09 01 = Id 265 = Gemälde. Diese Angabe finden wir direkt hinter der Angabe des Kamins, strukturiert man die Zeile um, so wird erkennbar, dass das Führungsbyte angibt, wieviele Item-Datensätze auf diesem Feld liegen. In unserem Fall sind es drei Items, die Mauer, der Kamin und das Bild. Da sie zusammen einen Satz bilden, benötigen die Itemangaben des Kamins und des Bildes kein Führungsbyte. Daraus ergibt sich eine abweichende Gesamtlänge gegenüber der Länge von drei Einzeldatensätzen. Jetzt ist die Struktur klar:

Führungsbyte / Doppelbyte Item / FF 01 im Falle, das nur ein Item auf der Fläche liegt
Führungsbyte / Doppelbyte Item / FF 01 / Doppelbyte Item / FF 01 .... bei mehreren Items.

Ein unbelegtes Feld dagegen wird einfach mit einem Null-Byte gekennzeichnet. Exemplarisch habe ich nun die vorletzte Spalte nach diesem Muster umgebrochen.
Die beiden Folgebytes FF 01, die jeder Itemangabe folgen, könnten sich auf die Anzahl der Items gleicher Art beziehen. Daraus lässt sich das Phänomen ableiten, warum man im Spiel, wenn man zunächst [Anzahl A] items und darüber [Anzahl B] items ablegt, nicht in einem Zug [Anzahl A+B] items aufnehmen kann. Der Server fügt bei jeder Ablage wie in einem Stack die abgelegten Items an, d.h. die oberste Schicht [Anzahl B] muss erst abgetragen sein, bevor man die darunter liegenden Items [Anzahl A] aufnehmen kann. (Prinzip LIFO, last in first out).

Ein Datensatz für die Itemablage auf einem Feld kann also
1 Byte, 5 Byte oder 5 Byte + (Führungsbyte -1) * 4 Byte lang sein.

So, das war etwas umfangreich, aber im Ergebnis doch völlig nachvollziehbar.

Aufbau einer *_container-Datei

Die Dritte im Bunde der Map-Dateien ist die kürzeste von allen. Hier der Hex-Dump:

```
12 00 14 00 5F 00 63 00 01 00 00 00 00 00
```

Ziehen wir unseren Header ab, so bleiben gerade mal vier Bytes übrig, und das sind auch noch Nullbytes. Das verwundert, wissen wir doch, dass sich auf der Map ein Depot befindet.

Schauen wir uns deshalb mal noch andere *_container-Dateien an :

Die größte zusammenhängende Map wird von der Ebene 0 gebildet. Der Ursprung liegt bei X -25 / Y-23, die Ausdehnung entnehmen wir wieder dem Header des Hexdump:

```
40 01 03 01 E7 FF E9 FF 00 00    ( 320 x 259 Felder )

36 00
00 00 F1 FF 9E 00 01
00 00 70 00 93 00 01
00 00 93 00 D7 00 01
00 00 C9 00 C8 00 01
00 00 70 00 8B 00 00
ED 00 0A 00 01 00 00
50 00 4F 00 01 00 00
94 00 78 00 01 00 00
6F 00 93 00 01 00 00
93 00 D8 00 01 00 00
96 00 D8 00 02 00 00
01 00 03 00 80 00 01
00 00 94 00 DC 00 01
00 00 91 00 79 00 01
00 00 98 00 24 00 01
00 00 97 00 DC 00 01
00 00 94 00 79 00 01
00 00 96 00 28 00 01
00 00 7B 00 93 00 01
00 00 93 00 D9 00 01
00 00 FB FF 98 00 01
00 00 96 00 D9 00 01
00 00 71 00 8B 00 00
61 00 78 00 01 00 00
03 00 81 00 01 00 00
60 00 59 00 01 00 00
95 00 78 00 01 00 00
```

```
96 00 29 00 01 00 00
93 00 DA 00 01 00 00
73 00 93 00 01 00 00
97 00 D8 00 02 00 00
01 00 96 00 24 00 01
00 00 71 00 8C 00 01
00 00 95 00 DC 00 01
00 00 99 00 24 00 01
00 00 95 00 79 00 01
00 00 DF 00 1D 00 01
00 00 7C 00 93 00 01
00 00 EC 00 0A 00 01
00 00 93 00 DB 00 01
00 00 4F 00 4F 00 01
00 00 96 00 25 00 01
00 00 93 00 D6 00 01
00 00 F5 FF F0 FF 01
00 00 EF FF B7 00 00
71 00 93 00 01 00 00
BE 00 4D 00 01 00 00
6F 00 8C 00 01 00 00
93 00 DC 00 01 00 00
90 00 79 00 01 00 00
97 00 24 00 01 00 00
96 00 26 00 01 00 00
96 00 DC 00 01 00 00
9A 00 24 00 01 00 00
```

Dem Header folgen 380 Bytes.
Die ID des Depot h (41 01) suchen wir vergebens.

Das Führungsbyte h36 könnte für 54 Datensätze stehen. Bei 7 Bytes pro Datensatz ergäben sich 7 x 54 = 378 Bytes. In der „leeren“ container-Datei zuvor waren dem Header vier Bytes gefolgt.

Nehmen wir für das Führungsbyte ein Doppelbyte, blieben zwei Byte übrig.
380 Bytes - 2 Bytes = 378 Bytes. Ein Datensatz ist also exakt 7 Bytes lang, das Führungsbyte beschreibt die Anzahl der Datensätze.

Das zweite und dritte Doppelbyte eines Satzes erscheint auch in der uns bereits bekannten Version, die an anderer Stelle als Negativwert erkannt wurde. Negativwerte kennen wir bislang nur aus dem Koordinatenbereich. Sind das zweite und dritte Paar eine Ortsangabe?

Außerdem wissen wir, dass die Ebene 0 keine 54 Depots enthält. Was also wird hier überhaupt dargestellt?

Wir greifen drei dieser Doppelpaare heraus und überprüfen ingame, was sich an diesen Stellen auf der Karte befindet:

```
0A 00 01 00    10 / 1         liegt mitten im Wasser
```

96 00 25 00 150 / 37

DC 00 01 00 220 / 1

Bereits nach Prüfung des ersten Satzes wird klar, dass die ausgewählte Angabe keinesfalls die Lage eines Depots darstellt. Auch eine andere Art von Behältnis, immer von der bezeichnung „container“ ausgehend, scheidet aus.

Zum jetzigen Zeitpunkt erscheint es mir möglich, dass diese Dateien zwar bei der Anlage einer Map eingerichtet werden, tatsächlich jedoch keine Funktion haben. So etwas kennen wir bereits im Zusammenhang mit den Dateien der Datenbank, wo etliche tables, bzw. einzelne colums keine Funktion haben.

Illarion_warpfields

Die Datei beinhaltet die Angaben über die Sprungpunkte im Spiel. Der entsprechende GM-Befehl kennt nur die Zielkoordinaten. Als Quellkoordinaten gilt der Standpunkt des GM zum Zeitpunkt der Aktivierung des Befehls.

37 00

3F 00 | 7A 00 | 01 00 | 3F 00 | 7A 00 | 00 00

FC 00 | DA 00 | 00 00 | 6D 00 | 72 00 | 00 00

usw

```
12 00 62 00 00 00 12 00 64 00 FD FF
E6 00 0C 00 01 00 E6 00 0C 00 00 00
38 00 72 00 01 00 38 00 72 00 02 00
38 00 72 00 02 00 38 00 72 00 01 00
62 00 64 00 00 00 62 00 64 00 01 00
32 00 2B 00 00 00 F3 FF 04 00 FD FF
62 00 64 00 01 00 62 00 64 00 00 00
5B 00 82 00 01 00 5B 00 82 00 00 00
12 00 63 00 FD FF 12 00 62 00 00 00
08 00 55 00 00 00 05 00 57 00 01 00
63 00 73 00 FD FF 77 00 81 00 00 00
96 00 33 00 00 00 12 00 07 00 FD FF
13 00 62 00 00 00 13 00 63 00 01 00
A4 00 DF 00 00 00 A4 00 DF 00 01 00
A4 00 DF 00 01 00 A4 00 DF 00 00 00
C8 00 0B 00 00 00 FB FF 3C 00 FD FF
63 00 64 00 00 00 C8 00 C8 00 00 00
12 00 07 00 FD FF 96 00 33 00 00 00
62 00 66 00 00 00 60 00 65 00 FD FF
05 00 57 00 01 00 08 00 56 00 00 00
A4 00 E0 00 FF FF A4 00 E0 00 00 00
A4 00 E0 00 00 00 A4 00 E0 00 FF FF
```

```
EF FF 05 00 00 00 11 01 79 00 00 00
85 00 60 00 00 00 85 00 60 00 01 00
42 00 72 00 01 00 42 00 72 00 02 00
42 00 72 00 02 00 42 00 72 00 01 00
FF FF 60 00 FD FF FF FF 60 00 00 00
FF FF 60 00 00 00 FF FF 60 00 FD FF
60 00 65 00 FD FF 62 00 66 00 00 00
00 00 D5 00 00 00 33 00 A3 00 00 00
0C 00 45 00 FD FF 0D 01 DA 00 00 00
11 00 62 00 00 00 13 00 63 00 01 00
FB FF 3C 00 FD FF C8 00 0B 00 00 00
77 00 81 00 00 00 63 00 73 00 FD FF
F6 00 23 00 FD FF F6 00 23 00 00 00
13 00 64 00 01 00 13 00 64 00 00 00
F6 00 23 00 00 00 F6 00 23 00 FD FF
03 00 C5 00 FD FF 03 00 C5 00 00 00
F3 FF 04 00 FD FF 32 00 2B 00 00 00
85 00 61 00 01 00 85 00 61 00 00 00
03 00 C5 00 00 00 03 00 C5 00 FD FF
03 00 C5 00 01 00 03 00 C5 00 02 00
03 00 C5 00 02 00 03 00 C5 00 01 00
E6 00 0B 00 00 00 E6 00 0B 00 01 00
64 00 64 00 01 00 64 00 64 00 02 00
64 00 64 00 02 00 64 00 64 00 01 00
F0 FF 05 00 00 00 11 01 79 00 00 00
6F 00 81 00 00 00 6F 00 81 00 01 00
6F 00 81 00 01 00 6F 00 81 00 00 00
03 00 C6 00 00 00 03 00 C6 00 01 00
03 00 C6 00 01 00 03 00 C6 00 00 00
5B 00 81 00 00 00 5B 00 81 00 01 00
3F 00 7A 00 00 00 3F 00 7A 00 01 00
```

662 Bytes, davon 2 Byte als Header (Anzahl Datensätze)
h37 = 55 Datensätze = 12 Byte pro Datensatz

Zu erwarten wären 2 x 2 Byte für Ausgangspunkt, 2 x 2 Byte für Zielpunkt, sowie 2 x 2 Byte für die Ebenen. Das sind 12 Byte.

Die Werte der ersten und vierten Spalte sind als Ebenenangabe ungeeignet. Daraus ergibt sich folgende Reihenfolge:

[von X] [von Y] [von Z] [nach X] [nach Y] [nach Z]

Nochmal erwähnenswert ist der Umstand, dass diese Datei nicht während des laufenden Servers manuell bearbeitet werden kann. Beim nächsten ordnungsgemäßen Serverdown würden die Alt-Daten des Arbeitsspeichers die Neueintragungen überschreiben.
Beim Löschen eines Datensatzes ist auch unbedingt darauf zu achten, dass der Header mit den Angaben zur Datensatzanzahl angepasst wird. Nach dem Rückschreiben der Daten auf den Server müssen unbedingt die Rechte angepasst werden, damit der Server beim anschließenden Neustart die Daten einlesen und bei erneutem Down zurückschreiben kann.

Die Anpassung ist also nicht schwierig, sondern „nur“ mit einigen Tücken verbunden. Das gilt im Übrigen analog für alle anderen Dateien des Map-Komplexes.

Illarion_specialfields

```
18 00

59 00 85 00 00 00 02 03 00 00 00
59 00 85 00 01 00 02 03 00 00 00
   85 00 73 00 00 00 02 03 00 00 00
      18 00 0F 00 FD FF 02 02 00 00 00
         64 00 76 00 00 00 02 03 00 00 00
            73 00 8B 00 00 00 02 03 00 00 00
                                 32 00 2B 00 00 00 02 03 00 00 00
                                 63 00 73 00 FD FF 02 04 00 00 00
      18 00 10 00 FD FF 02 04 00 00 00
                                 96 00 33 00 00 00 02 03 00 00 00
                                 C8 00 0B 00 00 00 02 03 00 00 00
                                 12 00 07 00 FD FF 02 04 00 00 00
                                 62 00 66 00 00 00 02 00 00 00 00
                                 60 00 65 00 FD FF 02 04 00 00 00
               52 00 75 00 00 00 02 02 00 00 00
                                 FB FF 3C 00 FD FF 02 04 00 00 00
                                 77 00 81 00 00 00 02 05 00 00 00
                                 F3 FF 04 00 FD FF 02 04 00 00 00
59 00 84 00 00 00 02 02 00 00 00
59 00 84 00 01 00 02 02 00 00 00
   85 00 72 00 00 00 02 02 00 00 00
               52 00 76 00 00 00 02 03 00 00 00
         64 00 75 00 00 00 02 00 00 00 00
            73 00 8A 00 00 00 02 05 00 00 00
```

Der Header weist 24 Datensätze aus. 264/24 ergibt 11 Bytes pro Datensatz.
Auffällig sind die häufigen Wiederholungen einzelner Angaben.
Bei den ersten vier Bytes handelt es sich wohl um die Koordinaten.
Die folgenden beiden Bytes sollten die Ebene klassifizieren.

In der bereits erwähnten Diplomschrift zum Projekt werden die Specialfields als Flächen deklariert, die dazu dienen, durch Fenster und Türen sehen zu können. Eine Überprüfung der Koordinaten müsste also stets einen Zusammenhang ergeben mit Fenstern und Türen.