

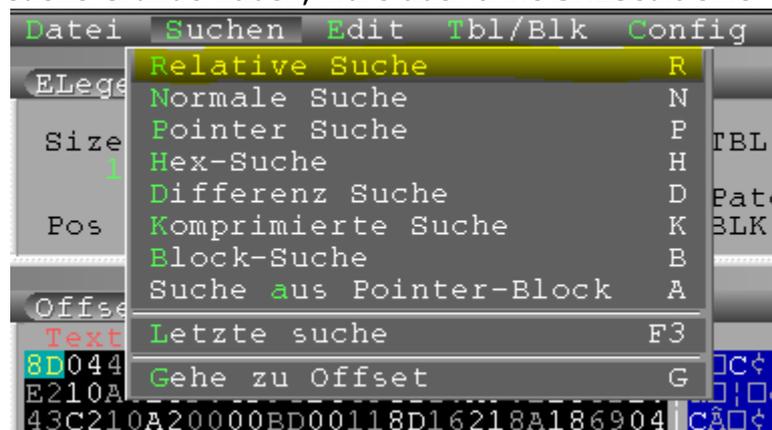
Hallo,

diese Anleitung wird darauf eingehen wie Zeichen in Spiele aufgerufen werden und wie man den dafür nötigen Table versteht und anwendet. Dazu kommt eine Erläuterung zum Hacken von unsortierten Alphabet-Systemen und japanischen Zeichensystemen.

Zuerst das Grundsätzliche: Jedes Zeichen in einen PC oder einer Konsole hat einen Wert. Daran orientiert man sich weitestgehend an die Reihenfolge im Alphabet:

A=00 ; B=01 ; C=02 usw.

Die zweistellige Schreibweise ergibt sich aus der Tatsache dass beim Hexeditor mit Hexwerten gearbeitet wird. Das bedeutet für uns dass der Buchstabe K den Wert 0A hat. Weiteres zum Hexadezimalsystem ist auf Wikipedia nachzulesen. Ein Hexeditor wird nun versuchen der Übersicht wegen die Hexwerte mit ASCII Zeichen anzuzeigen. Dabei muss aber das ASCII Alphabet nicht deckungsgleich mit dem ROM Alphabet sein. Das kann verschiedene Gründe haben, wäre aber ohne SNESedit eine Qual. SNESedit bietet uns die Möglichkeit anhand eines Textes eine Suche zu starten, die ein Alphabet ausfindig macht. Diese Suche heißt „Relative Suche“ und ist eines der wichtigsten Werkzeuge.



Ein unsortiertes Alphabet-System ist im Allgemeinen ein Alphabet welches nicht nach dem üblichen System ABCD... aufgebaut sondern ggf. nach Häufigkeit der Buchstaben sortiert ist.

Ein Alphabet im ROM wird genauso wie im echten Leben aufgezählt: A->1; B->2 usw. Das witzige dabei ist, bei einem japanischen Spiel bzw. einen mit unsortierten Alphabet ist das genauso. Wenn wir also die Zeichen im Spiel mit einen „normal“ sortierten Alphabet überschreiben entsteht im Spiel ein totaler Kauderwelsch. Ihr gebt Unordnung also eine Ordnung, mit der SNESedit etwas anfangen kann.

Zu aller erst suchen wir in unserem Spiel die japanischen Zeichen im Tileeditor. Wir beginnen angefangen beim ersten Zeichen, unsere lateinischen Buchstaben drüber zu "kleben". Dabei müsst ihr lediglich die Reihenfolge richtig halten und nur Groß-ODER Kleinbuchstaben verwenden.

In der Regel sind japanische Zeichen mit mehr Zeichen bestückt als unseres Alphabet hergibt. Diese ignorieren wir und lassen sie unbenutzt, vorerst. Wenn Sonderzeichen dazwischen sind, müsst ihr auch die mit überschreiben.

Die veränderten Werte speichern wir als IPS oder in den ROM um uns das Ergebnis anzuschauen.

Im Spiel wird jetzt ein Kauderwelsch zwischen japanischen und lateinischen Buchstaben sein. Das bringt uns weiter...

Wir gehen in die relative Suche und geben den Kauderwelsch ein. Dabei lassen wir die japanischen Zeichen als Sterne einfach zwischen den Buchstaben stehen. Dadurch dass wir eine Reihenfolge haben, die SNESEDIT bekannt ist. Können wir mit dieser Suche unseren Font und seine Werte ermitteln.

Die Restlichen Zeichen können nun für Sonderzeichen oder Ähnliches verwendet werden.

Tables verstehen und verwenden

Jeder Buchstabe und jede Ziffer hat einen Wert. Genau genommen einen Hexwert also 00 bis FF. Dieser Wert ist in den Konsolenspielen aber nicht standardisiert so, dass ein jedes Spiel einen anderen Wert haben kann für zum Beispiel den Buchstaben A.

Ein Table ersetzt dir den Wert oder viel mehr das Ergebnis in ein Sichtbares Alphabet. Um diese "Dolmetscherdatei" zu finden/bzw. zu erstellen, müssen wir eine relative Suche starten. Relativ deswegen weil wir nur den Text im Spiel, sowie die Ordnung des Alphabets kennen. Da die meisten englischen Spiele sowieso nur Großbuchstaben haben, sollte man teilweise schon mit "HELLO" oder "ENEMY" Erfolg haben.

SNESEDIT wird dich über die Suche an das Wort bringen wo es die Buchstabenkette gefunden hat. Nun kannst du mit dem drücken der Taste 1 direkt eine Table Datei erstellen.

Soweit so gut. Für viele Spiele sollte das genügen. Allerdings gibt es noch ein paar Sachen zu beachten.

Spiele können aus verschiedenen Gründen verschiedene Schriften haben. Schriften(Fonts) haben meist auch einen unterschiedlichen Buchstabenwert. Das heißt wir brauchen für jede Schriftart einen eigenen Table. Mit der Taste TAB(links von Q) kann man in Snesedit die Tables, aktivieren und auswählen.

Infos zum Hacken von Japanischen Zeichen folgt später.