Dieses Tutorial wird erklären wie die Positionen von Grafiken lokalisiert und verändert werden können.
Als Beispiel werde ich Metal Marines nutzen.
Weiterhin brauchen wir ein VRAM Viewer bzw. einen Emulator(mit Debugger).
Ich nutze in diesem Fall: VSNES und ZSNES. VSNES zeigt uns relativ einfach den VRAM bereits in den richtigen Einstellungen und wir können die Zeilen und Spaltenanzahl kontrollieren. Worauf ich aber später erst eingehe. Der VRAM sind die Grafiken und Schriftarten die die Konsole in den Grafik-Arbeitsspeicher legt.

Technisches Grundwissen:
Bildgrafiken werden nicht anders wie Textgrafiken geladen. Das heißt: Jede Grafik hat ihren eigenen HEX-Wert. Ihre eigene Adresse sozusagen. Zu diesen HEX-Wert kommt noch
ein Byte an Palleten-information dazu. Mehrere Grafiken hintereinander mit der selben Palette, können hierbei aber auch den Paletten Wert weglassen. Dieser wird dann am Anfang oder Ende stehen. Zumindest wenn der Programmierer diese Funktion genutzt hat. Wir gehen allerdings hier von Metal Marines aus und dort war der Programmierer nicht so speicherbewusst.

Ein kleines Beispiel dazu:



Ein Sprite hat die Maße 8 x 8 Pixel so zumindest beim SNES. Also haben wir 4 einzelne Sprites.

10 11 12 13 14



Nehmen wir an der erste Sprite hat im ROM den Wert 10 ; Der Zweite 11 ; und der Dritte 12 ; dazu der Vierte 13 und der letzte 14.
Weiterhin fantasieren wir uns den Farbwert 20 dazu.
Das würde also im Hexeditor so aussehen:

1020 1120 1220 1320 1420

Das Spiel macht also nichts anderes als folgendes: Es liest Zeilenweise die Grafiken sammt Farbpalleten-Wert dazu.
Sozusagen ist der HUD im Spiel nix weiteres als im ROM eine klare Linie von Adressen mit Werten für Farbe, in einer einfachen zeilenbasierenden Reihenfolge. Bei der Suche haben wir nur ein Problem: wir kennen in der Regel den Paletten- und Adressenwert nicht!

Den Wert des Tiles kann man aber ausrechnen:
Hierzu kommt unser VRAM-Viewer, VSNES zum Einsatz. Wir gehen mit ZSNES an den Punkt im Spiel wo ein HUD zusehen ist und machen einen Spielstand.
Diesen Spielstand laden wir jetzt in VSNES und gehen dort unter RAM Viewer.

Unser Screenshot



Unserer VRAM mit 4-bit



Unserer VRAM mit 2-bit



Mit ein bisschen rumspielen und ausprobieren werdet ihr die hier gebrauchten Funktionen von VSNES nutzen können.

Wichtig

Eure Grafiken müssen wenn ihr sie gefunden habt, in einer 16 Reihe pro Zeile ordnen. Sonst bekommt ihr keine Hexwerte heraus. Das Ganze sieht dann so aus:



Nun schnappt euch ein Zeichenprogramm eurer Wahl, macht einen Screenshot vom RAM und bastelt euch eine Übersicht dazu um nicht mühsam mit der Hand abzählen zu müssen.
Das ganze sieht dann aus wie Schiffeversenken.



Vergesst nicht dass man hier bei 0 anfängt mit Zählen.
So sieht es dann übersichtlich aus:

Der Wert nach unten ist der Erste, der Wert nach rechts der Zweite bei der Suche. Das große, dicke M hat also den Hexwert: 2A

Ich hoffe bis hierher ist soweit Alles klar.

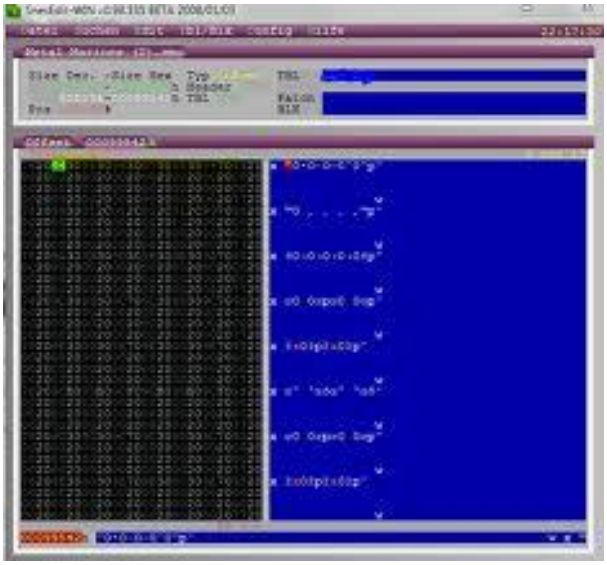
Nun zum Eingemachten. Wir wissen nun welchen Hexwert das Bild hat, ohne Palettenwert. Das stört und aber nicht. Wir können SNESEDIT einen HEX-Wert mit Relativen Zeichen machen. Ein relatives Zeichen ist das * (Das Mal Zeichen bei der nummerischen Tastatur).
Abgelesen von unserer Übersicht suchen wir also nach folgendem:

94**95**96**97**98**

Das entspricht diesen Tiles:
94 95 96 97 98



Nach ein bisschen Suchen, müsstet ihr das hier finden:



Wir haben jetzt die Grafikadressen gefunden, oder vielmehr die "Pointer" dafür. Dazu sogar die Farbwerte bzw. für die Spiegelung. Wir können jetzt also ohne Probleme andere Zeichen aufrufen lassen.
Dazu ändern wir einfach die Hexwerte(Taste X bei SNESEDIT) auf das gewünschte Tile und das Spiel wird es ohne Ärger reinladen. Hierbei ist unsere angefertigte Übersicht hilfreich.

Ich hoffe ich konnte euch Wissen vermitteln. Anregungen und Wünsche oder Fragen bitte im Forum posten.
Anbei möchte ich RED-Scorpion für seine Hilfe danken, dank ihm konnte ich mir das Wissen aneignen.

Hier eine vorgearbeitete Grafik zum Alphabet einfügen(Gimp):
http://black.bplaced.net/web/pics/others…ectsde/alph.xcf