

Tutorial zum Einfügen neuer Einheiten (Units) in ein bereits bestehendes Spiel

Diesem Tutorial setze ich voraus, dass das Packen und Entpacken der entsprechenden *.sue, *.aps oder *.pak Dateien nicht noch mal näher erklärt werden muss.

Auf das Einfügen von Bildmaterial zu diesem Tutorial habe ich hier bewusst verzichtet, um unnötige Verwirrungen zu vermeiden.

Ich möchte in meinem Tutorial auf das Einfügen von zwei Einheitentypen näher eingehen. Dies ist zum einen das Einfügen eines Fahrzeugs (auch Pak, MG...etc.), und zum Anderen das Einfügen einer ich nenne es mal „Person“ (damit meine ich Infanterie, Zivilist, oder auch Tier). Zwischen beiden Einheiten Typen besteht ein erheblicher Unterschied, was sich selbst an den Dateinamen, und deren Dateiendungen bemerkbar macht. Zum besseren Verständnis werde ich hier mal an zwei Beispielen erklären, welche Dateien dazu benötigt werden, und wo man diese findet. Weiter unten kommen wir dann zum Einfügen etc.

Fahrzeug:

Einfaches Fahrzeug-Beispiel aus SSRW: → M26Pershing Panzer

Die für das Spiel an sich relevanten Bilddateien findet man in der game_common(*.sue), und sie lauten in diesem Fall wie folgt:

M26pers.col	= color Palette (Farbpalette)
M26pers.hot	= diese Datei beinhaltet die Informationen, wo (an welchem Punkt) an diesem Fahrzeug die Schussanimationen sichtbar sein sollen, und an welchem Punkt sich der Turret (der Drehpunkt des Geschützturms) befinden.
M26pers.inf	= dies ist eine reine Informationsdatei in welcher definiert ist um welchen Fahrzeugtyp es sich hierbei handelt, hier kann z.B. auch die Anzahl der aufzusitzenden Soldaten festgelegt werden, und einiges mehr. Alle in der *.inf festgehaltenen Informationen haben direkten Einfluss auf die *.hot Datei. Dies wird aber in einem späteren Thread näher erklärt.
M26pers.pck	= dies ist die Datei mit den eigentlichen Bilddateien

Für den Editor, gibt es im Ordner Editor auch eine Datei namens edit_common(*.sue).

In dieser findet man die für den Editor wichtigen Dateien des o.g. Fahrzeugs welche da heißen:

M26pers-.col	= Bedeutung siehe oben
M26pers-.pck	= Bedeutung siehe oben

Mehr wird für den Editor nicht benötigt, da ja der Editor eben „nur“ die Einheit anzeigen soll, damit diese nach Wunsch gesetzt werden kann, und mehr nicht.

So, das waren jetzt die Bilddateien, was jetzt noch benötigt wird, ist natürlich auch die entsprechende Unit-Datei, in der sämtliche ich nenne es mal „Verhaltensregeln“ des Fahrzeugs (Fahrgeschwindigkeit, Explosion, Schussparameter usw.) festgehalten sind. Diese Datei trägt in unserem Fall den Namen pershing, und ist in dem Unterordner Units der desc_common(*.sue) zu finden. Hier befindet sich ebenfalls eine Datei namens units in der sämtliche Einheiten des Spiels aufgelistet sind. Dies ist für unser Vorhaben auch eine sehr wichtige Datei, doch dazu später mehr.

Person (Soldaten, Zivilisten, Tiere):

Einfaches Person-Beispiel aus SSRW: → amerikanischer Medic

Auch hier findet man die für das Spiel relevanten Dateien in der game_common, doch diese lauten aber jetzt etwas anders:

ALRUSMED.rs2 = die eigentliche Bilddatei (allerdings animiert!!)
SOLD.PL = dies ist die Farbpalette für alle im Spiel befindlichen *.rs2 Dateien

Ebenfalls werden auch hier wieder entsprechende Dateien für den Editor benötigt, welche wir wie oben schon erwähnt in der edit_common finden, und in unserem Fall wie folgt lauten:

ALLRUSMED.edt = Bilddatei für den Editor
SOLD.PL = Bedeutung siehe oben

Wie bei den Fahrzeugen, wird auch bei den Infanterie-Einheiten eine Unit-Datei benötigt, in der deren Verhalten (Laufgeschwindigkeit, Minen suchen/legen, Schussparameter ... etc.) festgehalten sind. Im vorliegenden Beispiel trägt unsere gesuchte Datei den Namen amedic und ist wie die Fahrzeug-Dateien ebenfalls in dem Unterordner Units der desc_common zu finden.

Sicherlich ist dem Einen oder Anderen jetzt aufgefallen, was das für eigenartige Dateinamen sind!? Während meiner doch langjährigen Beschäftigung mit diesem Thema bin ich immer wieder auf solch eigenartige Probleme gestoßen (dass Einheiten die merkwürdigsten Namen tragen), deshalb sehr schwer in der game_common und auch edit_common zu finden sind. Doch dieses Problem habe ich relativ einfach gelöst:

Tipp: Editor des Spiels starten aus dem man die Einheiten haben will, und die Einheiten im laufenden Editor suchen. Wenn die entsprechende Einheit gefunden ist einfach den Namen der Einheit, welcher in Klammern hinter der Einheit steht notieren. Dies ist der Name der Einheit der definitiv auch in dem Unterordner Units der desc_common zu finden ist, und schon hat man die Unit-Datei, in welcher man dann sehen kann welchen *.rs2-Namen, und welchen *.edt-Namen unsere gesuchten Einheiten tragen. Die Sold.pl spielt dabei keine Rolle, da sie bei allen gleich ist.

Beispiel (Auszug) aus der Unit-Datei des amedic:

File ALRUSMED

***picture ALRUSMED.edt SOLD.pl**

Die Bezeichnung file gibt den Namen der *.rs2 Datei in der game_common an, und die Bezeichnung *picture ALRUSMED.edt SOLD.pl gibt die Bilddatei (zuzüglich der Farbpalette) in der edit_common an. Alle Informationen aus den Unit-Dateien, welche mit einem * beginnen beziehen sich übrigens nur auf den Editor!

So, jetzt wo wir wissen, welche Dateien benötigt werden und wo und wie wir diese finden können kann der eigentliche Spaß ja losgehen. Das Vorgehen ist, egal um welche Modifikationen es sich handelt, grundsätzlich immer dasselbe.

Wir gehen in unserem Beispiel mal von einem Spiel aus welches auch fast jeder kennt → den TGM-Mod. In diesem finden sich viele faszinierende Einheiten, welche man gerne mal in SSRW oder

sonstige integrieren möchte. Auf speziellen Wunsch gehe ich zum Einen mal auf den Arzt, und zum Anderen mal auf einen Patton Panzer ein, welche im TGM-Mod zu finden sind. Diese möchte ich in mein vorhandenes SSRW einfügen, und gehe dabei wie folgt vor:

Als erstes habe ich mal den Editor des TGM-Spiels gestartet, um herauszufinden welche genauen Bezeichnungen (Unit-Namen) meine gesuchten Einheiten tragen, um diese dann in der Units-Datei zu finden. Nach einigem Suchen im Editor habe ich meine gewünschten Einheiten gefunden, und mir deren Units-Namen notiert, welche da heißen:

civ_M.sld = für den Arzt

M48.tnk = für den Patton Panzer

Nun kann ich den Editor schließen, und mich meinen bereits vorher separat entpackten TGM-Spieldateien widmen.

In TGM ist die Ordnerstruktur etwas anders wie bei SSRW, doch das soll uns jetzt nicht abschrecken. Der TGM beinhaltet keine game_common sondern halt eine game_data, welche aber letztendlich dieselben Daten beinhaltet welche von uns benötigt werden. Nur dass hier eben die Einheiten-Dateien (Bilddateien fürs Spiel) und auch einige Animationsdateien direkt mit vorhanden sind. Egal, zweimal hingesehen, und schon hat man den Überblick gefunden.

Nun öffnen wir den Ordner UNITS und suchen nach den beiden von uns notierten Dateien (civ_M.sld und M48.tnk). Um irgendwelche Katastrophen beim kopieren zu vermeiden, sollte man sich (zumindest mache ich es immer so) auf dem Desktop für jede neue Einheit einen Ordner erstellen. In diesem Ordner wiederum 3 Unterordner erstellen. Diese sollten dann in game, edit und units benannt werden. Damit die kopierten Dateien letztendlich nicht durcheinander geraten, und auch relativ schnell wieder an den richtigen Platz im neuen Spiel gelangen können.

Nachdem wir uns jetzt für unseren Panzer und für unseren Arzt jeweils einen Ordner nebst Unterordnern zugelegt haben kopieren wir uns die Datei civ_m.sld in den Unterordner units des Arzt Ordners. Gleiches machen wir mit der Datei M48.tnk, nur halt in den Unterordner units des Patton Ordners. So, die Units-Dateien haben wir schon mal. Nun öffnen wir die jeweilige Units-Datei um heraus zu finden welche Namen unsere entsprechenden Bilddateien für das Spiel (game_common) und für den Editor (edit_common) haben.

In unseren Beispielen wären das für den Arzt (civ_M.sld):

File civ_M

***picture civ_M.edt sold.pl**

Das heißt:

civ_M.rs2, und civ_M.edt brauchen wir für unser Vorhaben. Die sold.pl kann vernachlässigt (nicht beachtet) werden da diese in unserem SSRW Game bereits vorhanden ist.

Und für den Patton (M48.tnk) wären das:

File M48a5

***picture M48a5-.pck M48a5-.col**

Das heißt :

Wir brauchen jetzt zum Einen die Dateien M48a5-.pck und M48a5-.col für unsere edit_common, und zum Anderen alle!! M48a5 Dateien für unsere game_common. An dieser Stelle entstehen beim Modden die meisten Fehler, denn hier wird meistens die Hälfte vergessen!!!

Also Wichtig! Alle Dateien namens M48a5, welche wie folgt sind:

M48a5.pck ; M48a5.col ; M48a5.hot und M48a5.inf

Dies sind jetzt die Dateien, die wir für unsere game_common benötigen.

Nun geht's wieder ans Kopieren, und das heißt die Datei civ_M.rs2 muss in den Unterordner game des Arzt-Ordners, und die Datei civ_M.edt muss in den Unterordner edit des Arzt Ordners kopiert werden.

Gleichmaßen gehen wir mit den Dateien des Patton Panzers vor. Die Dateien mit der -.pck und -.col Endung (dieses „Minus“ kennzeichnet immer Editordateien) müssen in den Unterordner edit des Patton Ordners, und die Dateien M48a5.pck, M48a5.col, M48a5.hot und M48a5.inf müssen in den Unterordner des Patton Ordners kopiert werden.

OK, kurzer Zwischenstand:

Wir haben jetzt zwei Ordner einen Arzt und einen Patton Ordner auf unserem Desktop. Beide Ordner beinhalten jetzt jeweils einen units-Ordner (units-datei) für unsere desc_common einen game Ordner mit den Bilddateien für unsere game_common sowie einen edit Ordner mit den Bilddateien für unsere edit_common. Alle o.g. Dateien haben wir aus dem TGM in diese Ordner kopiert! Dann können alle Ordner des TGM-Mods erst einmal geschlossen werden, und wir öffnen unsere SSRW Ordner wo wir ja die neuen Einheiten einfügen wollen.

Das Einfügen der neuen Einheiten:

Nun, da wir ja clever genug waren uns separate Ordner nebst Unterordnern anzulegen, haben wir jetzt leichtes Spiel. Jetzt einfach die Dateien aus unseren angelegten Ordnern in die entsprechenden Ordner des SSRW Spiels hineinkopieren.

Das heißt:

Die Dateien aus den units Ordnern in den Ordner UNITS der desc_common von SSRW kopieren.
Die Dateien aus den game Ordnern in den Ordner game_common von SSRW kopieren.
Die Dateien aus den edit Ordnern in den Ordner edit_common von SSRW kopieren.

Achtung !! Sehr wichtig ist, wie ich bereits weiter oben schon geschrieben habe, die units Datei im Ordner UNITS. Hier sind ja alle Einheiten für SSRW aufgelistet, und auf diese Liste greift die Spiel-Engine zu! Das wiederum bedeutet, wenn wir neue Einheiten hinzugefügt haben, dann müssen diese Einheiten auch hier eingetragen werden! In unseren Beispielen haben wir es relativ einfach, da im Dateinamen (Endung) bereits darauf hingewiesen wird, um welche Typen es sich handelt.

M48.tnk (tnk = Tank) ; civ_M.sld (sld = Guner)

Tipp: Sollten die Dateiendungen mal nicht vorhanden sein, dann einfach in der entsprechenden units-Datei des Spiels (aus dem man die Datei entnehmen möchte)nachsehen um welchen Typ es sich handelt, und diesen dann im neuen Game in der units-Datei des Ordners UNITS eintragen.

Nun haben wir die neuen Einheiten in unser Spiel integriert, und jetzt bleibt nur noch das Anpassen des „Verhaltens“ (siehe oben) und der entsprechenden Sounds, Shots, Explosions und Animationen.

Dazu gibt es zwei Möglichkeiten:

Möglichkeit 1 → man sucht sich im Spiel „gleichwertige/gleichartige“ Einheiten-Dateien und übernimmt die Daten (Parameter) aus der entsprechenden units-Datei in die neue Datei. Man muss hier lediglich darauf achten, dass zum Einen die Namen und Bezeichnungen der neuen Einheit (rs2, edt, pck, col) beibehalten werden, und zum Anderen sollte auf die „Gleichwertigkeit“ geachtet werden. Auch hier folgt gleich ein genaues Beispiel!

Möglichkeit2 → man muss schauen, ob die in der units-Datei geforderten Sounds, Shots, Animations und Explosionsdateien im SSRW vorhanden sind. Wenn nicht, dann müssen diese halt Schritt für Schritt aus dem TGM-Spiel ebenfalls herauskopiert und ins SSRW Spiel eingefügt werden. Dies ist eigentlich auch keine schwierige Sache, da die „meisten“ Daten in sämtlichen Sudden Strike Games vorhanden sind. Also, wenn etwas fehlt dann ist es nicht viel. Man muss halt nur darauf achten, wenn bei Shots, Explosions und Animats zusätzliche Dateien hinzugefügt werden, das diese auch in den „Listings“ der Ordner eingetragen werden. Diese „Listings“ findet man wie folgt:

Animats = im Ordner Animats

EXPLOSIONS = im Ordner Explosions

SHOTS = im Ordner Shots

Zu den Sounds, folgt in Kürze ein separates Tutorial!

Nun, nochmal zu unseren Beispielen, damit die Dinger ans Laufen kommen wenden wir uns erstmal der Möglichkeit 1 (gleichwertige units-Datei) zu:

Für unseren „Arzt“ bietet sich ja geradezu an, die units-Datei von unserem (in der Einleitung erwähnten) amedic zu benutzen. Dazu einfach im UNITS Ordner (SSRW) der desc_common eine Kopie von amedic erstellen. Diese Kopie in civ_M.sld umbenennen. Dann die Datei öffnen und die Einträge Name, File und *picture entsprechend der Original civ_M.sld abändern.

Nun sollte unser Arzt im Spiel wie auch im Editor seine Verwendung finden, da ja alle anderen relevanten Daten (Sounds, Animats etc.) im Spiel eh vorhanden sind.

Für unseren Patton Panzer kann man sich einen ebenbürtigen Panzer raussuchen und im Grunde genommen dasselbe wie mit dem Arzt machen. Beispiel (ok vielleicht a Little Bit to strong) der Pershing Panzer! Die units-Datei pershing kopieren, die Kopie in M48.tnk umbenennen, und die Einträge Name, File und *picture entsprechend der Original m48.tnk anpassen, fertig!

Auch der Panzer kann jetzt über den Acker rollen!

Wie gesagt, wer's ganz genau haben will, der sollte dann schon auf Shots, Explosions, Animats und Sounds achten, doch mit ein wenig üben hat man schnell den Bogen raus.

So, ich hoffe ich habe einigen geholfen, dieses Thema wenigstens ansatzweise zu verstehen. Für Fragen und Anregungen stehe ich gern zur Verfügung.

Ich wünsche Euch viel Erfolg!

Medevac

