Plattformkompatibilität

Inhaltsverzeichnis

- 1 Dateinamen
- 2 Umlaute
- 3 Speicherformate

Erfreulicherweise läuft Train Fever nicht nur auf Windows, wie die meisten Spiele, sondern es wurde auch an die Nutzer von Linux und Mac gedacht. Speziell beim Erstellen von Mods kann dieser an sich positive Umstand aber zu Problemen führen, da die Systeme nicht 100%ig kompatibel sind, was den Austausch von Dateien untereinander betrifft. Ich bin selber darüber gestolpert als ich ein Mod von der Webdisk installieren wollte, im Verlauf der Diskussion hat sich dann herausgestellt das ich nicht der Einzige mit diesem Problem bin. Zweck dieser Sammlung ist einerseits den Usern Tips und Ansatzpunkte zu geben wenn ein Mod nicht so funktioniert wie es soll. Andererseits liest vielleicht der ein oder andere Modder dies und achtet dann ein wenig darauf. Ich beziehe mich hier insbesondere auf Linux und Windows. Zu Mac kann ich nicht viel sagen, da ich keinen habe. Hinweise und Ergänzungen - vor allem diesbezüglich - sind natürlich willkommen.

Welche Stolpersteine gibt es also beim Austausch von Dateien?
Im Wesentlichen drei:

- Dateinamen
- Sonderzeichen und Umlaute
- Speicherformat

1 Dateinamen

Windows ignoriert die Groß- und Kleinschreibung bei Dateinamen. Die Dateien haben zwar *scheinbar* unterschiedliche Dateinamen, intern speichert sie das zum Einsatz kommende Dateisystem jedoch komplett kleingeschrieben ab, Daraus resultiert auch die Unmöglichkeit, in einem Verzeichnis Dateien unterzubringen, die den gleichen, aber unterschiedlich geschriebenen Namen haben: Test.txt, TEST.TXT und test.txt können nicht gemeinsam in einem Verzeichnis existieren. Unter Linux hingegen ist das möglich, ebenso beim Mac.

Ein Beispiel:
Ein Modder erstellt ein neues Fahrzeug. Er referenziert in seinem Mod eine Datei "radkasten_vorn.mdl". Die tatsächliche Datei heißt jedoch "RadKasten_Vorn.mdl" Nun passiert folgendes:
Windows: "radkasten_vorn.mdl" und "RadKasten_Vorn.mdl" sind quasi das Gleiche. Das Spiel wird das mod anstandslos laden.
Linux & Mac: Die Groß- und Kleinschreibung wird beachtet. Das Mod wird nicht geladen und - meiner Erfahrung nach - schmiert das Spiel mit einer Fehlermeldung ab.

Tipp dazu: Auf korrekte Groß- und Kleinschreibung achten und sicherstellen das die nachgeladenen Dateien exakt den Namen haben mit dem sie in der Mod referenziert werden. Am einfachsten ist es hier einfach alles klein zu schreiben.

2 Umlaute

Grundsätzlich sind Umlaute auf keiner der Plattformen auf denen Train Fever läuft ein Problem. Ein Fettnäpfchen entsteht erst, wenn die Dateien zwischen unterschiedlichen Systemen ausgetauscht werden. Der Grund dafür liegt darin, das jedes der Systeme seine eigene Art und Weise hat, mit den Umlauten umzugehen. Nachdem der ursprüngliche ASCII Zeichensatz keine Umlaute bot, hat man diesen erweitert, um es Nutzern zu ermöglichen in ihrer eigenen Sprache zu schreiben ohne irgendwelche Verrenkungen zu machen. Heute gibt es eine Vielzahl an Zeichensätzen. Beispielsweise wäre ISO8859-1 zu nennen, das ist der lange Zeit in Westeuropa übliche Zeichensatz. Dessen Nachfolger ist UTF8. UTF8 ist ein neuer Standard, hier werden für jedes Zeichen, anstatt dem bisherigen einen Byte, wenigstens zwei Bytes (Schriften mit lateinischen Buchstaben) und bis zu vier Byte (Schriftarten mit nicht-lateinischen Buchstaben wie chinesisch, japanisch, arabisch etc) verwendet. Darüber hinaus haben viele Hersteller eigene Zeichensätze entwickelt oder bestehende Zeichensätze erweitert und diese dann zusammen mit ihrer Software vermarktet. Da nun die unterschiedlichen Plattformen die unterschiedlichen Zeichensätze nicht immer verstehen, wissen sie gegebenenfalls nicht wie sie mit den Daten umgehen sollen und interpretieren sie falsch. Dann wird zum Beispiel aus "grün" ein "grÃ¼n" oder aus "Räder" wird ein "RÃ¤der". Oder schlimmeres.

Tipp dazu: Um dieses Problem gänzlich zu vermeiden, ist es am Besten vollständig auf Umlaute und sonstige Sonderzeichen zu verzichten und nur den ursprünglichen ASCII Zeichensatz zu verwenden. Müssen aus irgendeinem Grund dennoch Umlaute sein, so muss man darauf achten, die Datei mit dem richtigen Zeichensatz zu speichern, so das ein anderes Programm erkennen kann, was ihm da vorgeworfen wurde. Soweit ich weiß, mag Train Fever als solches die Umlaute auch nicht gerade, insofern bietet sich das Weglassen durchaus an. Das bezieht sich allerdings auf die Programmstruktur / lua-Skripte und Dateinamen. Text innerhalb von Anführungszeichen oder innerhalb von Variablen (z.B. Beschreibung eines Fahrzeugs) ist davon nicht betroffen.

3 Speicherformate

Windows und Linux haben, aus historischen Gründen, unterschiedliche Arten eine Textdatei abzuspeichern. Genauer gesagt: Die enthaltenen Zeilenumbrüche an jedem Zeilenende. Windows (und MS-DOS) speichern dabei nach jeder Zeile ein "Carriage Return"-Zeichen sowie ein "Newline"-Zeichen. Linux hingegen speichert nur das "Newline"-Zeichen. Dies stammt noch aus der Zeit, in der es keine Bildschirme gab und Drucker, als einziges Ausgabegerät, explizite Anweisungen zum Bewegen des Druckkopfs bekommen mussten.
Öffnet man nun eine Datei die unter dem jeweils anderen Betriebssystem gespeichert wurde, steht gegebenenfalls der komplette Text in einer Zeile. Oder es sind lauter "merkwürdige" Zeichen am Ende jeder Zeile.

Tipp dazu: Dies ist ein Problem das man nicht völlig umgehen kann. Die meisten modernen Texteditoren sind mittlerweile in der Lage die Datei zu erkennen und schon beim Lesen und Speichern zu konvertieren, so das

der Benutzer gar nichts davon mitbekommt. Da man die Datei auch nur in einem der Formate speichern kann, hilft alles andere wenig. Für die Linuxbenutzer gibt es "dosunix", ein praktisches Kommandozeilentool das hin- und her konvertieren kann und in praktisch jeder Distribution enthalten ist.