

LCD Matrix Berlin

Lawo für MAN DL (X10) und Solaris GN (alterr)

Version 1.0
27.11.2016

Modell, Texturen, Skript: Teneberus
Schriften: Daniel Seiler & Teneberus

1 Einleitung

Hallo!

Wenn dich das Gelaber hier nicht interessiert, kannst du gleich zur **Installation** springen.

Noch da? Gut. Hier sind ein paar ausgewählte Informationen zur Anzeige aus diesem Paket.

Erst einmal vielen Dank an Daniel Seiler, der mich mit den Schriften versorgt hat. Ohne diese hätte ich die Anzeige wohl nie fertigstellen können.

Außerdem bedanke ich mich auch bei allen anderen, die mich unterstützt haben. Ich habe es nicht vergessen ;-)

Motivation

Es gibt ein paar Gründe dafür, dass ich die Matrix fast vollständig von Grund auf neu entwickelt habe. Bisher existierende Anzeigeskripte sind meistens für meine Zwecke völlig überladen. Die meisten Funktionen werden für die Berliner LCD Anzeige gar nicht benötigt. Die Anzeige soll ganz einfach nur das darstellen, was in der Hofdatei steht, das heißt: Beachtung von Groß- und Kleinschreibung und keine speziellen Formatierungen. Außerdem soll doch auch die originale Auflösung und zumindest annähernd die richtige Größe umgesetzt werden, wenn es schon ein reales Vorbild gibt. Das erfordert natürlich neue Modelle und Texturen.

Skriptaufbau

Einige Zeilen sind aus der originalen mit OMSI 2 ausgelieferten Krüger Matrix übernommen. Das sind die Funktionen für die KI und die Seiten- und Steckschilder. Alles andere, also die eigentliche Anzeige, ist vollständig neu geschrieben, funktioniert aber nach dem selben Prinzip:

1. Ermittlung von Liniennummer und Zieltext
2. Schriftwahl und Positionsberechnung
3. Schreiben der Daten auf eine „Speicherbitmap“, auf der ein Pixel einem Punkt auf der Anzeige entspricht.
4. Punktweise Vergrößerung und Übertragung dieser „Speicherbitmap“ auf die eigentliche Anzeigentextur.

Aus Gründen der Übersichtlichkeit und auch der Größe der Texturen, sind diese Schritte auf mehrere Frames aufgeteilt. Die ersten drei Schritte geschehen jeweils in einem Frame, der vierte benötigt vier Frames (einer pro Anzeige).

Textur

Die vom Skript verwendete Textur hat eine Größe von 1024x512px und beinhaltet vier verschiedene Anzeigengrößen. Ein Punkt ist dabei 5px² groß.

- 180x26 Punkte bzw. 5 Elemente mit jeweils 36x26 Punkten.
- 192x26 Punkte bzw. 4 Elemente mit jeweils 48x26 Punkten.
- 48x26 Punkte (das ist nur ein Element).
- 144x26 Punkte bzw. 3 Elemente mit jeweils 48x26 Punkten.

Fast alle Fahrzeuge, auch die beiden mit diesem Paket versorgten, verwenden nur die ersten drei Größen. Die vierte Größe ist für die Fahrzeuge mit stehendem Motor und breiter Heckanzeige (Beispiel: Berliner Citaro, Baujahr 2002) vorgesehen.

Das Skript beschreibt auch diesen Teil schon, obwohl er noch gar nicht benötigt wird. Das liegt daran, dass ich ein einziges Skript für alle Fahrzeuge haben wollte, sodass bei einer Aktualisierung nur eine Datei in alle Verzeichnisse kopiert werden muss.

Universelle Hofdatei

Wie bereits erwähnt, soll die Anzeige exakt das anzeigen, was in der Hofdatei steht. Es werden keinerlei Kodierungszeichen oder sonstige Spielereien beachtet. Außerdem wird die Groß- und Kleinschreibung einfach übernommen.

Damit die Ziele also genau so dargestellt werden, wie sie dargestellt werden sollen, sind zwingend neue Strings erforderlich. Keiner der bereits vorhandenen wird den Anforderungen gerecht. Diese neuen Strings tragen die Nummern 23 und 24 in der universellen Hofdatei. Aber keine Angst... Das Skript verwendet auch weiterhin die Strings 1 und 2, falls 23 und 24 nicht existieren oder leer sind. Wenn du damit leben kannst, dass die Ziele ausschließlich in Großbuchstaben dargestellt werden, kannst du also auch die alten Hofdateien weiterverwenden.


























Die neuen Strings unterliegen erstmal keinen Beschränkungen. Das Anzeigenskript überprüft jedoch nur die ersten 24 Zeichen auf Unterlänge. Längere Strings sind trotzdem möglich. Auswirkungen auf die vertikale Ausrichtung wird das nicht haben, da bei so einer Zeichenlänge sowieso die kleineren Schriften verwendet werden, bei denen das keinen Unterschied macht.

Für das X10 Addon liegen diesem Paket modifizierte Hofdateien bei. Dabei handelt es sich jedoch nicht um eine vollständige universelle Hofdatei. Es wurden lediglich die Strings 23 und 24 ergänzt.

Außerdem war es mir zu aufwendig, die Strings im alten Format hinzuzufügen. Leider kann man im neuen Format nicht einfach Teile der Hofdatei weglassen, weshalb die reduzierte Datei weiterhin alle Haltestellen und Routen enthält. Einen negativen Einfluss auf die Performance konnte ich bisher nicht feststellen.

Symbole

Es können selbstverständlich auch Symbole anstelle der Liniennummer dargestellt werden. Über das IBIS (im Solaris GN ohne ALMEX) sind das bisher nur ein Dreieck (Code: 2) oder „Hänsel&Gretel“ (Code: 3). Alle weiteren Symbole müssen über das OMSI-Menü eingestellt werden. Dazu muss im Feld für die Liniennummer folgendes eingegeben werden:

 – <code>\s{flowers}</code>	 – <code>\s{planeS}</code>	 – <code>\s{u}</code>	 – <code>\s{star}</code>	 – <code>\s{triangle}</code>
 – <code>\s{bbg}</code>	 – <code>\s{ball}</code>	 – <code>\s{wrench}</code>	 – <code>\s{loco}</code>	 – <code>\s{a7}</code>
 – <code>\s{runner}</code>	 – <code>\s{pr}</code>	 – <code>\s{trunk}</code>	 – <code>\s{coffee}</code>	 – <code>\s{db}</code>
 – <code>\s{notes}</code>	 – <code>\s{tram}</code>	 – <code>\s{tree}</code>	 – <code>\s{ice}</code>	 – <code>\s{bvg}</code>
 – <code>\s{hg}</code>	 – <code>\s{s}</code>	 – <code>\s{e}</code>	 – <code>\s{arrow}</code>	 – <code>\s{planeL}</code>

Kombinationen aus Symbolen und Zahlen sind bisher nicht möglich.

2 Installation

Die Installation der Anzeigen ist leider etwas aufwendiger. Da es sich bei den zu modifizierenden Fahrzeugen zum einen teilweise um Payware handelt und zum anderen jeder selbst die ein oder andere Modifikation durchgeführt hat, die ich nicht zunichte machen möchte, liefere ich keine „fertigen“ Dateien mit. Es muss also alles manuell erledigt werden.

Du solltest den grundlegenden Aufbau der *.cfg Dateien im Modellverzeichnis kennen. Wenn dir das fremd ist: Lerne es! Eine hoffentlich ausreichende Kurzübersicht gibt es [hier](#). Der Rest sollte auch ohne besondere Kenntnisse verständlich sein.

Die durchzuführenden Schritte sind erstmal bei jedem Fahrzeug identisch:

- 1.a Hinzufügen der Schriften (Das ist natürlich nur einmal nötig)
- 1.b Hinzufügen des Modells (*.o3d Datei), der Texturen und der Skriptdateien
- 2.a Austausch der Skript-Texturen und der Objekte in der Modellkonfiguration
- 2.b Austausch der Skriptdateien in der Buskonfiguration

Du musst natürlich nur die Anweisungen befolgen, die das Fahrzeug betreffen, das du mit der Anzeige ausrüsten möchtest.

Wichtig: Erstelle von jeder Datei, die du bearbeitest, mindestens eine Sicherungskopie (auch „Backup“ genannt), aber nicht im selben Verzeichnis!

Dateien hinzufügen

Die neuen Dateien liegen in diversen Unterverzeichnissen in diesem Paket.

Achtung: Die Verzeichnisstruktur entspricht **nicht** der von OMSI 2. Einige Dateien werden von allen Fahrzeugen benötigt und ich wollte diese nicht doppelt (oder sogar noch öfter) mitliefern.

Die **Schriften** befinden sich im Verzeichnis **fonts/** und müssen nach **<Installationspfad>/Fonts/** kopiert werden. Dabei ist der **<Installationspfad>** in der Regel etwas in der Richtung von **.../Steam/SteamLibrary/SteamApps/common/OMSI 2/**. Je nachdem, wo du Steam und OMSI 2 installiert hast.

Die **Modelle** befinden sich in den Verzeichnissen **model_dl/** (DL (X10)) und **model_sgn/** (Solaris GN). Das dortige Unterverzeichnis (**lcd_matrix**) muss in das Modellverzeichnis des jeweiligen Fahrzeugs kopiert werden.

Kopiere also

- den Inhalt von **model_dl/** nach **<Installationspfad>/Vehicles/MAN_DL05/Model/** und
- den Inhalt von **model_sgn/** nach **<Installationspfad>/Vehicles/Solaris BVG/Model/**.

Die **Texturen** beförderst du natürlich in das entsprechende Texturverzeichnis.

Kopiere also den Inhalt von **textures/**

- nach **<Installationspfad>/Vehicles/MAN_DL05/Texture/** für den DL (X10), bzw.
- nach **<Installationspfad>/Vehicles/Solaris BVG/Texture/** für den Solaris GN.

Als letztes sind die **Skriptdateien** dran. Dazu befinden sich im Verzeichnis **script/** zwei Unterverzeichnisse:

lcd_matrix_bvg_lawo/

 und **lcd_matrix_bvg_lawo_x10ai/**.

Für den DL (X10) werden beide Verzeichnisse, das heißt also wieder der ganze Inhalt von **script/**, nach **<Installationspfad>/Vehicles/MAN_DL05/Script/** kopiert.

Für den Solaris GN wird nur das Unterverzeichnis **lcd_matrix_bvg_lawo** nach **<Installationspfad>/Vehicles/Solaris BVG/Script/** kopiert.

Dateien modifizieren - Allgemeines

Nun kommt der nervige Teil: Das Modifizieren der verschiedenen Dateien. Du musst ein paar Abschnitte in den genannten Dateien durch neue „Bausteine“ ersetzen. Diese findest du im Unterverzeichnis **src/** dieses Pakets. Dort liegen verschiedene Textdateien für die einzelnen Dateien der Fahrzeuge. Ich werde in der folgenden Beschreibung auf diese Dateien verweisen.

Dieser Abschnitt ist nach Fahrzeugen aufgeteilt. Den Anfang macht der DL.

Dateien modifizieren - DL (X10)

Wechsel nun bitte in das Verzeichnis **<Installationspfad>/Vehicles/MAN_DL05/**. Dort befinden sich acht *.bus-Dateien - je eine für die Bauserien DL05, DL07, DL08 und DL09 sowie deren KI-Versionen. Im Vorbild besitzen nur die beiden letztgenannten die Lawo-Anzeigen, weshalb ich auch nur die Modifikation dieser beiden Serien beschreiben werde. Die anderen funktionieren jedoch identisch, so dass du dir auch diese mit der Anzeige ausrüsten kannst.

Öffne als erstes die beiden Dateien **MAN_DL08.bus** und **MAN_DL09.bus** und folge in diesen den Anweisungen aus der Datei **src/bus_dl.txt** dieses Pakets.

Als nächstes öffne die Dateien **MAN_DL08_KI.bus** und **MAN_DL09_KI.bus** und folge in diesen den Anweisungen aus der Datei **src/bus_dl_ki.txt** dieses Pakets.

Damit sind die Buskonfigurationen abgeschlossen. Der nächste Schritt betrifft drei Skriptdateien im Unterverzeichnis **Script/** des Fahrzeugs. Gemeint sind die Dateien

MAN_DL05_AI.osc,
MAN_DL05_KI_AI.osc
und MAN_DL05_KI_main.osc.

Erstelle von jeder Datei eine Kopie und nenne sie

MAN_DL05_AI_mod.osc,
MAN_DL05_KI_AI_mod.osc
und MAN_DL05_KI_main_mod.osc.

In jeder dieser Kopien musst du den Apostroph (') vor dem Eintrag (**M.L.Matrix_frame**) entfernen.

Als letztes sind die Modellkonfigurationen dran. Wechsel dazu bitte in das Unterverzeichnis **Model/** des Fahrzeugs. Wie schon im Hauptverzeichnis des Fahrzeugs, gibt es hier für jede Bauserie und deren KI-Versionen eine Datei.

Öffne die Dateien **DL_08.cfg**, **DL_08_KI.cfg**, **DL_09.cfg** und **DL_09_KI.cfg** und folge in diesen den Anweisungen aus der Datei **src/model_dl.txt** dieses Pakets.
(Hier sind die Kenntnisse dieser Dateien erforderlich ;-))

Hiermit ist die Installation der Anzeige im DL (X10) vollständig abgeschlossen.

Dateien modifizieren - Solaris GN

Wechsel bitte in das Verzeichnis **<Installationspfad>/Vehicles/Solaris BVG/**. Hier befinden sich vier *.bus-Dateien, die bearbeitet werden wollen.

Öffne zunächst die Dateien **Urbino 18 Main.bus** und **Urbino 18 Main_ALMEX.bus** und folge in diesen den Anweisungen aus der Datei **src/bus_sgn.txt** dieses Pakets.
Danach öffne die Dateien **Urbino 18 Trail.bus** und **Urbino 18 Trail_ALMEX.bus** und folge in diesen den Anweisungen aus der Datei **src/bus_sgn_trail.txt** dieses Pakets.

Zum Abschluss folgen jetzt noch einmal die Modellkonfigurationen. Wechsel dazu bitte in das Unterverzeichnis **Model/** des Fahrzeugs.

Öffne hier die Dateien **model_Urbino 18.cfg** und **model_Urbino 18_ALMEX.cfg** und folge in diesen den Anweisungen aus der Datei **src/model_sgn.txt** dieses Pakets.
Als letztes öffne die Datei **model_Urbino 18 Trail.cfg** und folge in dieser den Anweisungen aus der Datei **src/model_sgn_trail.txt** dieses Pakets.

Hiermit ist die Installation der Anzeige im Solaris GN vollständig abgeschlossen.

3 Lizenz

Die Veröffentlichung erfolgt unter der Lizenz CC BY-NC (<https://creativecommons.org/licenses/by-nc/3.0/de/>).

4 Anhang: Kurzeinweisung in Modellkonfigurationsdateien

Ich führe hier keine vollständige und detaillierte Aufklärung bezüglich dieser Konfigurationsdateien durch, sondern beschreibe nur rudimentär genau den Teil, der für die Installation der Anzeige notwendig ist. Für alle weiteren Informationen musst du andere Quellen konsultieren.

Zur erfolgreichen Durchführung der Änderungen in diesen Dateien sind zwei Abschnitte wichtig:

1. Skript-Texturen
2. Meshs

Beide werden durch entsprechende Schlüsselwörter eingeleitet und bestehen aus diversen Zeilen an Definitionen, die selbst wiederum durch Schlüsselwörter eingeleitet werden können.

Ein **Mesh** beginnt beispielsweise durch das naheliegende Schlüsselwort **[mesh]**. Der Einfachheit halber kann man sagen, dass alles, was darauf folgt, zu diesem Mesh gehört, bis das nächste Mal ein Schlüsselwort auftaucht, dass entweder ein neues Mesh oder eine andere Option einleitet. Bei den Meshs in dieser Anleitung sind das jedoch immer andere Meshs. Das heißt, dass du einfach das zu ersetzende Mesh suchst, das natürlich mit dem Schlüsselwort **[mesh]** beginnt. Dort entfernst du dieses **[mesh]** und alles bis zu der Stelle, an der das nächste Mal das Schlüsselwort **[mesh]** steht (aber ohne dieses) und fügst an derselben Stelle das neue Mesh ein, wenn es eins gibt.

Skript-Texturen sind einfacher. Diese werden durch **[scripttexture]** eingeleitet und bestehen nur aus zwei Zeilen, die die Auflösung der Textur angeben. Hier werden einfach alle Skript-Texturen (inkl. der Kommentare) gelöscht und durch die beiden neuen ersetzt.