

Anleitung zur Implementation von Recorded Messages und HTML Skript Anzeigen

Dateistruktur:

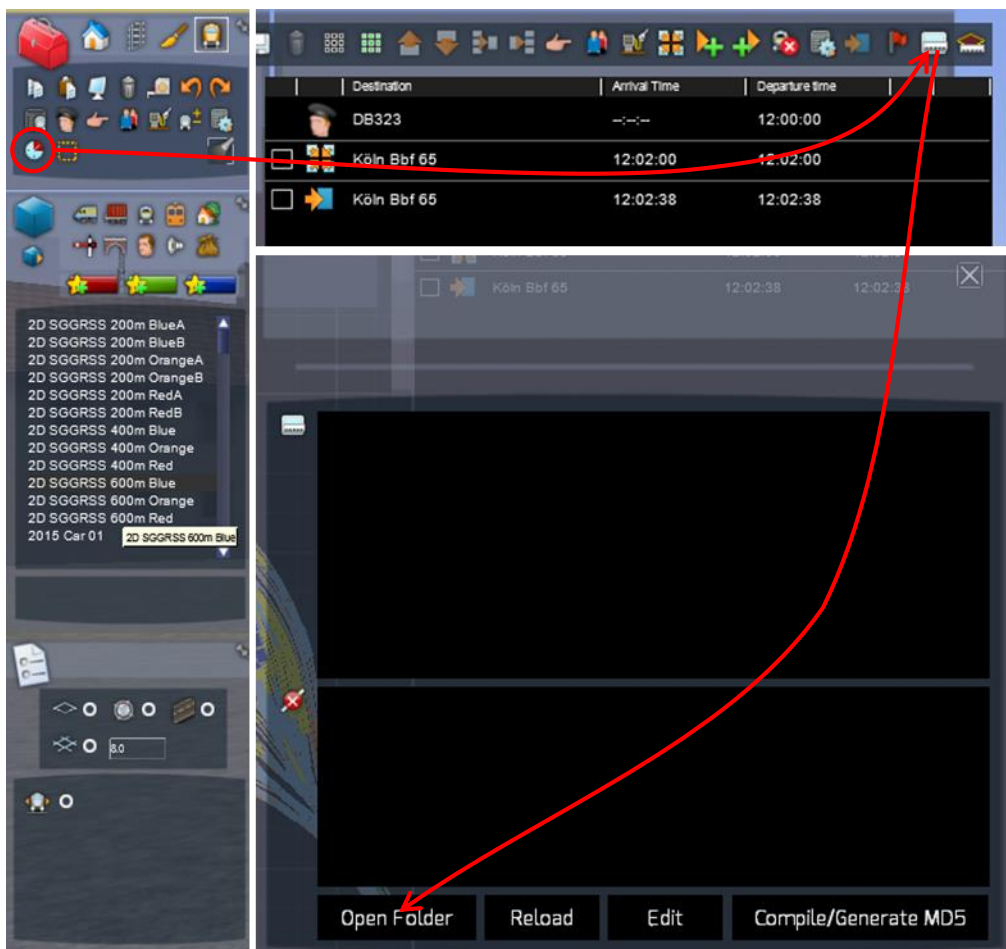
Wichtig ist die Dateistruktur beizubehalten, ansonsten kann Trainsimulator (TS) die Dateien nicht finden.

2 Wege zur Navigation in Explorer:

- 1) Über den TS Build Tab



- 2) Über den Scenario Builder



Der Explorer schaut so aus:

Name	Date modified	Type	Size
InitialSave.bin	12/6/2015 11:16 PM	VLC media file (.bi...	37 KB
InitialSave.bin.MD5	12/6/2015 11:16 PM	MD5 File	1 KB
Scenario.bin	12/6/2015 11:16 PM	VLC media file (.bi...	41 KB
Scenario.bin.MD5	12/6/2015 11:16 PM	MD5 File	1 KB
ScenarioProperties.xml	12/6/2015 11:16 PM	XML File	16 KB
ScenarioProperties.xml.MD5	12/6/2015 11:16 PM	MD5 File	1 KB

Hier erstellen wir jetzt Ordner, einen für jede Sprache, und die Datei ScenarioScript.lua. Wichtig die Datei ScenarioScript.lua muss im root-Verzeichnis des Scenarios liegen.

In diesem Beispiel sind die Ordner De für Deutsch, EN für Englisch, die ScenarioScript.lua und die Bilddatei DB-128x128.png.

Name	Date modified	Type	Size
De	19/6/2015 6:19 PM	File folder	
En	19/6/2015 6:19 PM	File folder	
DB-128x128.png	30/5/2015 6:05 PM	IrfanView PNG File	5 KB
InitialSave.bin	12/6/2015 11:16 PM	VLC media file (.bi...	37 KB
InitialSave.bin.MD5	12/6/2015 11:16 PM	MD5 File	1 KB
Scenario.bin	12/6/2015 11:16 PM	VLC media file (.bi...	41 KB
Scenario.bin.MD5	12/6/2015 11:16 PM	MD5 File	1 KB
ScenarioProperties.xml	12/6/2015 11:16 PM	XML File	16 KB
ScenarioProperties.xml.MD5	12/6/2015 11:16 PM	MD5 File	1 KB
ScenarioScript.lua	23/5/2015 3:40 PM	LUA File	3 KB

In den Sprachen Ordner sind momentan zwei HTML-Dateien:

Name	Date modified	Type	Size
exit.html	24/5/2015 1:00 AM	HTML File	1 KB
intro.html	19/6/2015 6:27 PM	HTML File	1 KB

Die HTML's

```
<html>
  <body>
    <table>
      <tr>
        <td>
          
        </td><td>
          <font color="#000000" face="Calibri" size="2">DB323</font>
          <br>
          <br>
          <font color="#D44A0B" face="Arial" size="5"><b>Fahrdienstleiter<br>ruft ...</b></font>
        </td>
      </tr>
    </table><table>
      <tr>
        <td>
          <font color="#000000" face="Arial" size="3"><p>Text</p></font>
        </td>
      </tr>
    </table>
  </body>
</html>
```

Unbedingt anzumerken ist das TS nur simple HTML's anzeigen kann und bilder nur 64x64 und vielfaches davon. In dieser HTML sind es zwei Tabellen, die erste mit zwei Zeilen und die zweite mit einer und jeweil unterschiedlich formatierter Text. Die erste Tabelle erste Zeile ruft das Bild ab, darum liegt es eine struktur weiter oben damit alle Sprachen auf die gleiche Datei zugreifen.

ScenarioScript.lua:

Ich verwende zur bearbeitung WSciTE: <http://www.scintilla.org/SciTEDownload.html>

Variablen deklarationen:

```
-- true/false defn
FALSE = 0
TRUE = 1

-- condition return values
CONDITION_NOT_YET_MET = 0
CONDITION_SUCCEEDED = 1
CONDITION_FAILED = 2

-- Message types
MT_INFO = 0 -- large centre screen pop up
MT_ALERT = 1 -- top right alert message

MSG_TOP = 1
MSG_VCENTRE = 2
MSG_BOTTOM = 4
MSG_LEFT = 8
MSG_CENTRE = 16
MSG_RIGHT = 32

MSG_SMALL = 0
MSG_REG = 1
MSG_LRG = 2
```

Die ersten beiden sind einfach nur True/False deklaration. Die nächsten drei für Szenario Beeinflussung, brauchen wir hier nicht. Alle übrigen sind Nachrichtentyp, Nachrichtenposition und Nachrichtengröße.

Recorded Messages:

```
-- Recorded Messages

-- function DisplayRecordedMessage( messageName )
    SysCall("RegisterRecordedMessage", "StartDisplay" .. messageName, "StopDisplay" .. messageName, 1);
end

-- function StartDisplayIntroText()
    SysCall("ScenarioManager:ShowInfoMessageExt", "INTERCOM", "intro.html", 0, MSG_TOP + MSG_LEFT, MSG_REG, TRUE );
end

-- function StopDisplayIntroText()
end

-- function StartDisplayExitText()
    SysCall("ScenarioManager:ShowInfoMessageExt", "INTERCOM", "exit.html", 0, MSG_TOP + MSG_LEFT, MSG_REG, TRUE );
end

-- function StopDisplayExitText()
end
```

Die Funktion Display Recorded Messages ist notwendig und speichert die Nachricht von Syscall, diese ist im Spiel wieder abrufbar. Die Syscall registriert dann die Nachricht mit einem Start „StartDisplay“ und Ende „StopDisplay“.

Jede Nachricht hat jeweils eine StartDisplay und eine StopDisplay auch wenn sie leer ist. In der StartDisplay rufen wird die HTML ab. Die Message Box heisst Intercom und lädt die intro.html. Danach steht die Anzeigendauer in Sekunden. Die 0 heisst die Nachricht eingeblendet wird bis der User das Kreuz drückt. Danach stehen die Textbox position und grösse. Am Schluss ist entweder TRUE, das Spiel wird unterbrochen oder FALSE das Spiel läuft während der Anzeige weiter.

Event Handler:

```
-- Events

-- function OnEvent(event)
    return _G["OnEvent" .. event]();
end

-- function OnEventIntroText()
    DisplayRecordedMessage("IntroText");
end

-- function OnEventExitText()
    DisplayRecordedMessage("ExitText");
end

-- Testconditions

-- function TestCondition(condition)
    return _G["TestCondition" .. event]();
end
```

Jede ScenarioScript.lua hat eine function OnEvent() und eine function TestCondition() zu enthalten auch wenn sie leer sind.

Dabei ist die OnEvent dafür einen event vom Spiel zu bearbeiten so wie hier bei Aufrufen von IntroText über DisplayRecordedMessages eine Nachricht anzuzeigen. Der name IntroText hier ist wichtig da wir damit den Event aus dem Spiel aufrufen. Das Konstrukt return_G erlaubt jeden Event in einer eigenen Funktion zu bearbeiten.

Mit der funktion TestCondition kann spielbeeinflussend sein. Mann kann damit die Geschwindigkeit abrufen, Beschleunigung abrufen, kontrollieren ob das Licht eingeschalten ist, sämtliche Kontrollen sperren usw.

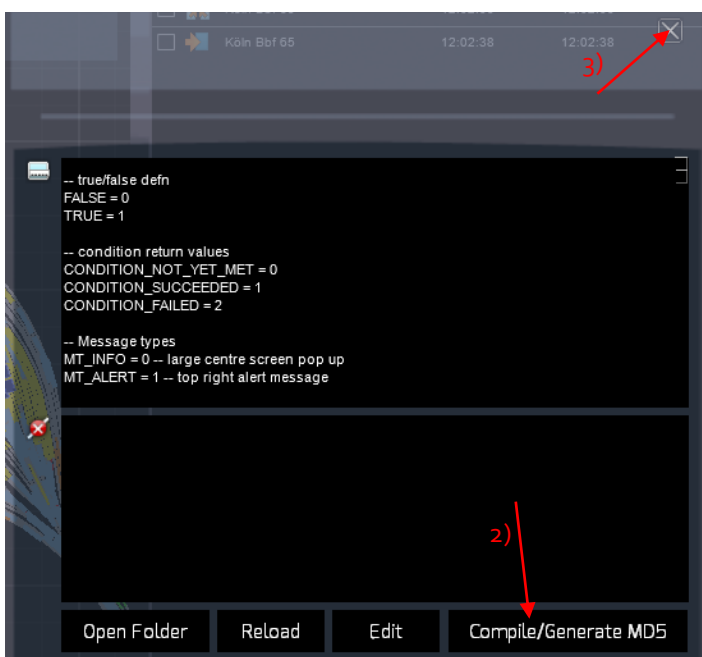
Wenn man das alles hat dann zurück in den Scenario Builder

Lua Aufruf im Scenario Builder

Als erstes müssen wir die ScenarioScript.lua in TS kompilieren:



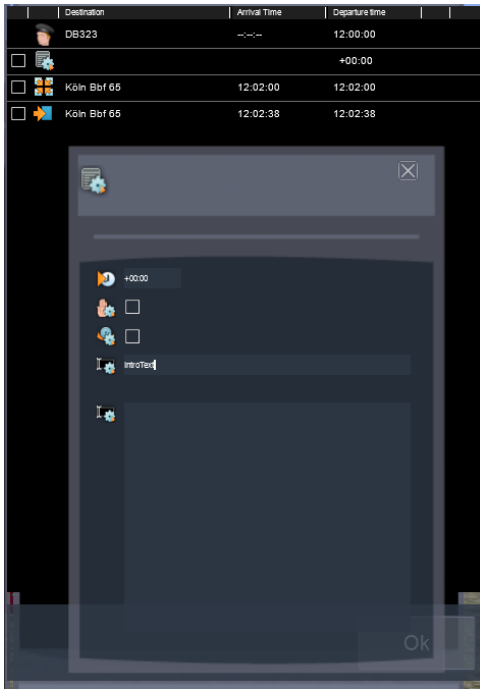
Klick Reload, wenn die ScenarioScript.lua am richtigen Ort ist und keinen Fehler enthält sollte das Bild so aussehen:



Das obere Textfeld sollte einen Ausschnitt von der ScenarioScript.lua enthalten. Das untere Textfeld muss leer sein, ansonsten ist ein programmier-fehler in der ScenarioScript.lua.

Wenn alles passt dann klick Compile/Generate MD5 und verlassen die Anwendung. Im Hintergrund hat TS einige Dateien in den Szenario Ordner geschrieben.

Jetzt zum Aufruf:



Ein einfacher Eventtriggerer mit dem Namen IntroText ist alles was man braucht.

Das Ergebnis sieht dann so aus:

