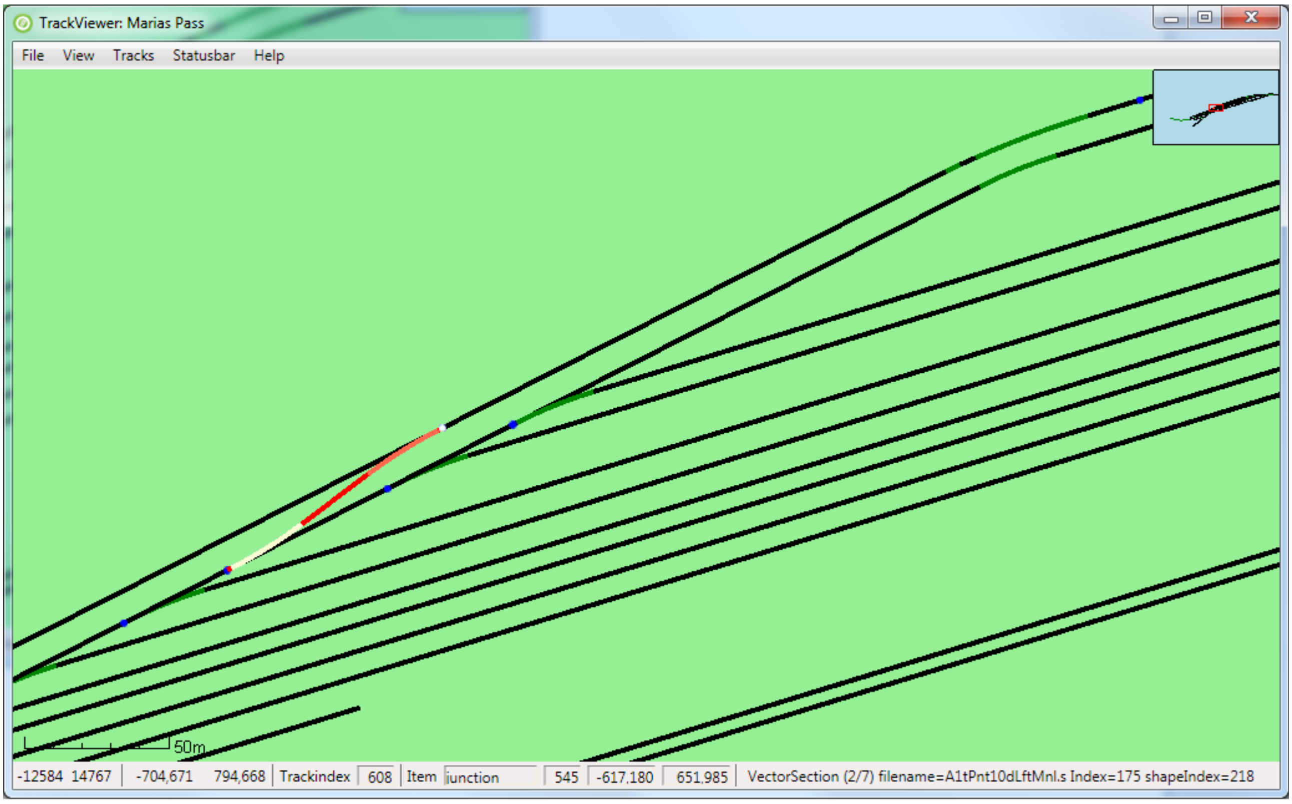
ORTS TrackViewer manual

Introduction

ORTS TrackViewer is an open source program to view tracks and all track items from a MSTS (Microsoft Trains Simulator) route. TrackViewer is very similar to the program MSTS TrackViewer, which is no longer developed. It does already have number of advantages over MSTS, e.g. the possibility to draw paths (from activities).



```
-12584  14767 | -704,671    794,668 | Trackindex  608 | Item  junction    545 | -617,180 | 651,985 | VectorSection (2/7) filename=A1tPnt10dLftMnl.s Index=175 shapeIndex=218
```

It is my hope that TrackViewer will become a part of ORTS, the Open Rails Train Simulator (www.openrails.org). It uses the source code of ORTS, especially to read all needed MSTS files and create the needed data-structures. At this point in time TrackViewer works without modifying the ORTS source code.

By using the ORTS code, TrackViewer is also a tool that can be used as a debugging tool for those parts of ORTS that directly deal with tracks and paths.

Installation

The current release contains source code only. And only the source code that is not already part of ORTS. This means that for now you can only run it when you are able to download and compile recent ORTS source code. For more information on this, see www.openrail.org/source.html.

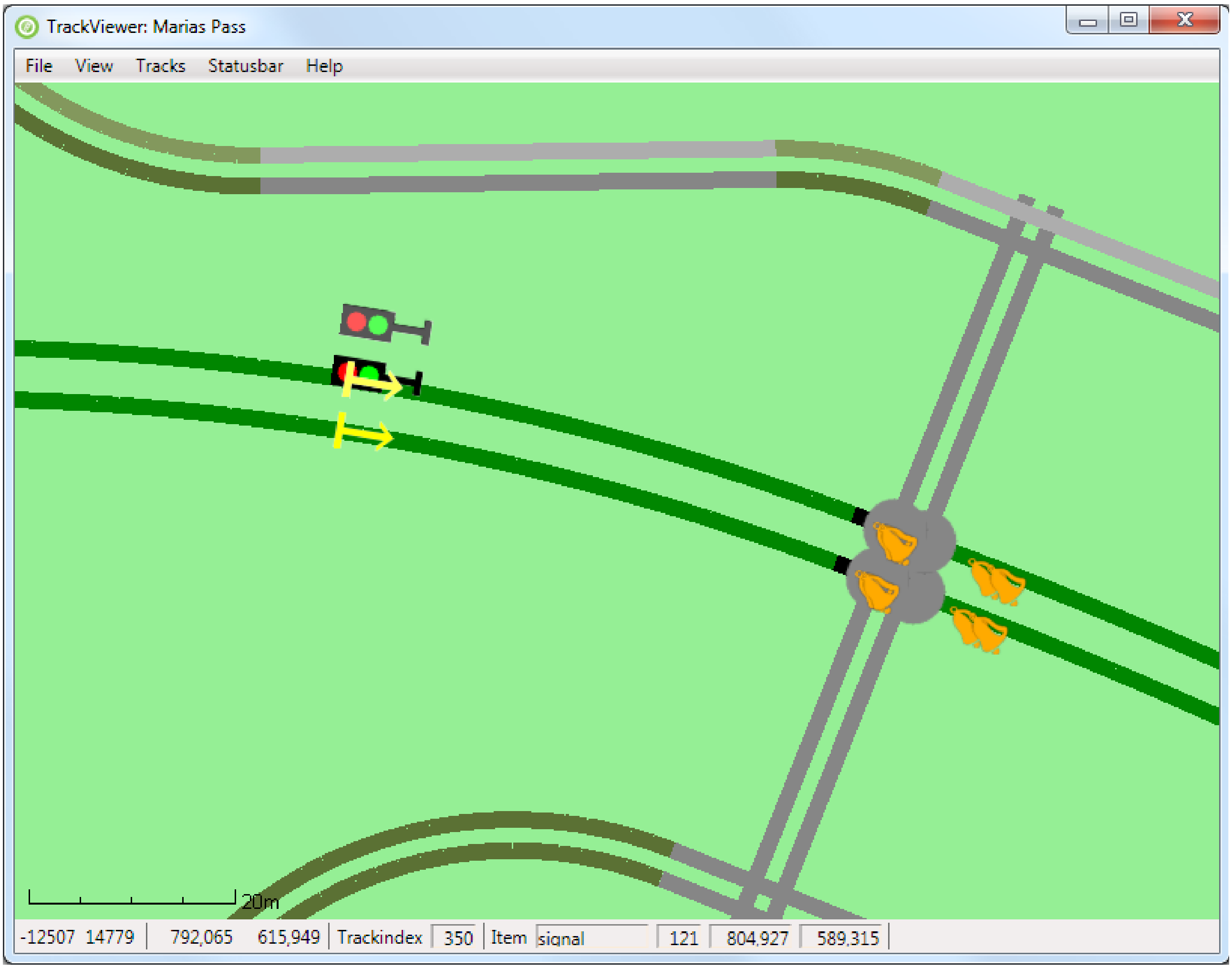
This code should not be very sensitive to updates on the ORTS code, simply because it only uses pretty stable parts of the ORTS code. Nevertheless, it does need a recent version of ORTS, X1966 or later.

To install TrackViewer you need to copy the directory TrackViewer to the source directory of ORTS as a separate project. This means it needs to be a directory next to the directories Launcher, Menu, RunActivity and others. The next step is to add the project to the ORTS visual C# solution. To do this, go to the 'Solution Explorer', click the right-mouse button, Add, Existing project, and then select the TrackViewer.csproj. If everything is fine, you can now compile the solution. This will generate an executable TrackViewer.exe in the Program directory, so next to OpenRails.exe, Menu.exe and RunActivity.exe. TrackViewer.exe does depend on RunActivity.exe, so it cannot run standalone.

Manual

When running TrackViewer for the first time, it will try to find your MSTS installation. If it does not find it (or in case you did not install MSTS), you can select the install directory under the File menu. You can then load a route, again via the File menu. Selecting a different route can also be done via the File menu. After the first use, you can reload your previous route (this was saved).

Most of the menu items are pretty self-explaining. You can select which items and which tracks to draw.

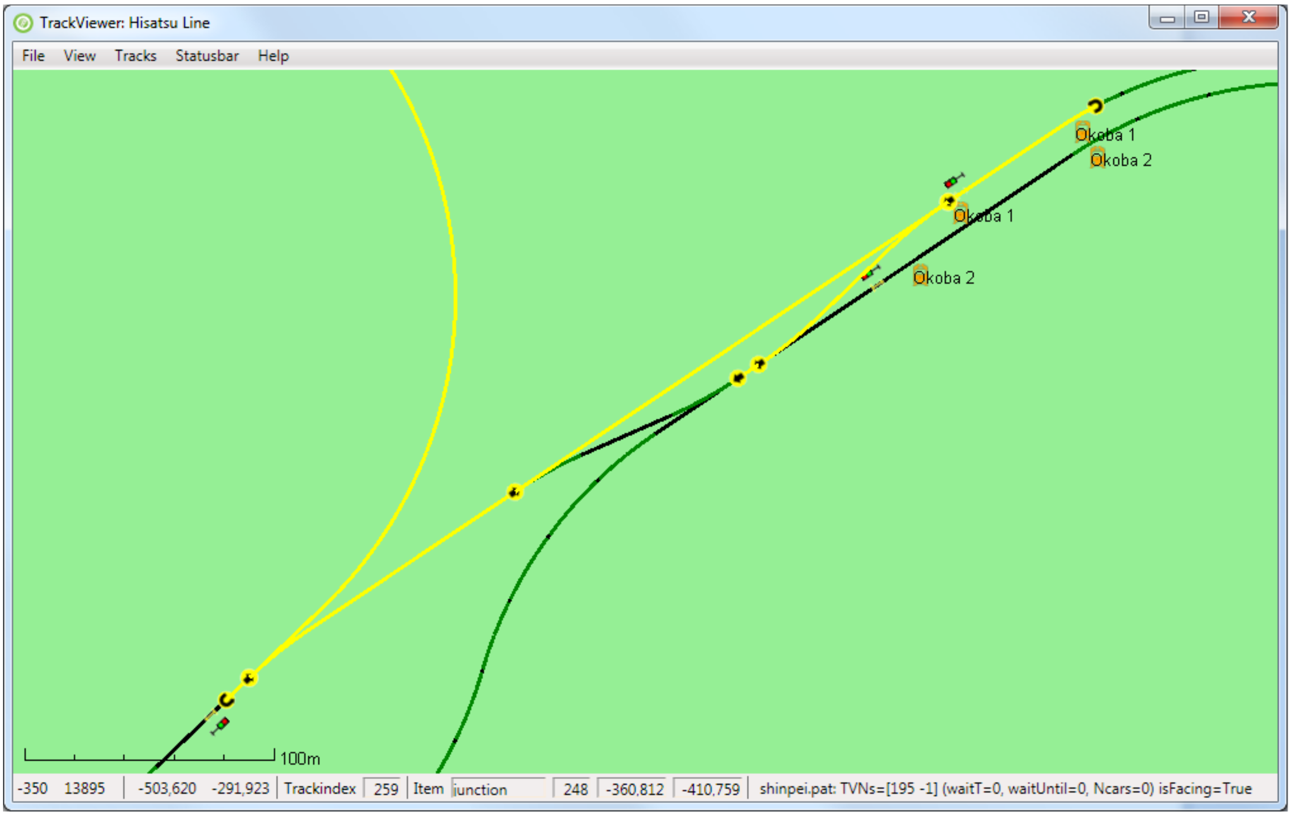


```
-12507  14779 | 792,065   615,949 | Trackindex | 350 | Item | signal | 121 | 804,927 | 589,315
```

TrackViewer will keep most of your settings. This includes the install directory, the last route, and what you selected to be drawn on screen and what not. In other words, almost all selections work as preferences at the same time.

A number of menu items have keyboard shortcuts (also mentioned on the menu items). Other shortcuts can be found under Help. Zooming and shifting works best using plus and minus keys ('=' and '-') or the mouse wheel. Pressing shift during zooming with the mouse wheel gives you finer control. Pressing shift Zooming is centered around the mouse position normally. Shifting the view window can be done using the keys a-s-d-w, or with the mouse (left-mouse button pressed).

TrackViewer also supports the drawing of paths, as used by activities for both player train and AI trains. These paths are defined in the PATHS directory of the route and have extension .pat. Drawing can be done in two ways. Drawing the raw .pat file itself will draw a very crude path, containing only straight lines. Drawing 'processed path' is the more advanced feature and uses the ORTS AIPath code. Initially only the first part of the path is drawn. The length of the drawn path can be decreased/increased with the PageDown and PageUp keys. Shift-PageUp will draw the full path, Shift-PageDown will only draw the starting point. Using the key 'c' will center the view window on the last drawn point of the path. It can be kept pressed during zooming or during the decreasing/increasing the path length.



-350 13895 | -503,620 -291,923 | Trackindex 259 | Item junction 248 -360,812 -410,759 | shinpei.pat: TVNs=[195 -1] (waitT=0, waitUntil=0, Ncars=0) isFacing=True

Releases

2014-02-02

Initial release to Elvas Tower.

2014-02-11

Release with a number of fixes and additions based on user feedback

- Added endnodes
- Added capability to search for tracknodes and items
- Route is no longer loaded immediately. This gives the possibility to change menu items and make it easier to recover from errors.
- Route names with non-ascii characters now supported and should no longer crash.
- Added option to select background color.
- Added discrete zooming levels (only a fixed number of possibilities exists for the amount of meters per pixel). This makes it easier to use the same zoom level as before.

    - Finer control on zooming by using shift while using the mouse scroll-wheel.

Future development

Any future development depends on the wishes and needs of the community. I created ORTS TrackViewer as a debugging tool initially (working on paths), and it grew into something much more.

The following items have already been requested. Some of these might end up being implemented

- Make a path editor for MSTS paths
    - apparently including grade crossings would be a benefit compared to MSTS.
- Currently it is possible to search for tracknodes and trackitems
    - Possibly it would be nice to have problems in the route directly available from the viewer (instead of getting this information from OpenRailsLog.txt).
- Add possibility to import and export routes. So people can have a look at a route without having it installed.
    - Currently this is already possible by copying global tsection.dat, route-specific tsection.dat, <route>.tdb (and <route>.rdb), and for the moment also <route>.trk.
    - Making an export/import routine would basically mean to make a different file format to write and read (at least part of) the information in these files. It is not clear whether this is worth the effort.
- Make TrackViewer independent of XNA. This prevents people to have install XNA. Currently this is quite a big change, and it would probably also need the code to be independent of ORTS itself.

Feb 11th, 2014, Jeroen.