FPS Display/Measurement/Analysis

Inhaltsverzeichnis

- 1 Introduction
- 2 Display of external programs
- 3 Debug Window (Rendering)
- 4 FPS Monitor
- 5 Script vs Graphic Performance
- 6 Influencing Factors
    - 6.1 External
    - 6.2 Transport Fever specific
    - 6.3 View
    - 6.4 Interaction/GUI
    - 6.5 Console
- 7 Analysis
- 8 Test ideas / Performance questions

1 Introduction

Performance is always a relevant topic at Transport Fever and mostly a problem that affects all systems. Simulation games like TPF in general can never get enough computing power and users always want to play on bigger maps with more and more vehicles and mods. TPF certainly has a few individual problems, although I assume that Urban Games is aware of the importance of this topic and also a lot has happened since Train Fever up to the last performance patch.

There are already numerous discussions on this topic, which deal with the effects of game settings, mods and individual hardware components on performance. The **FPS** is almost always used as an indicator for it and beside the average value, also the stability or (micro) stutters are of interest. It is therefore important to register them reliably.

Since manual reading is always afflicted with uncertainty, I would also like to introduce my Mod Tool **FPS-Monitor**. It makes the objective and automated measurement of the frame rate under the same environmental conditions possible. This allows to examine effects in more detail and performance influences can be detected more precisely.

2 Display of external programs

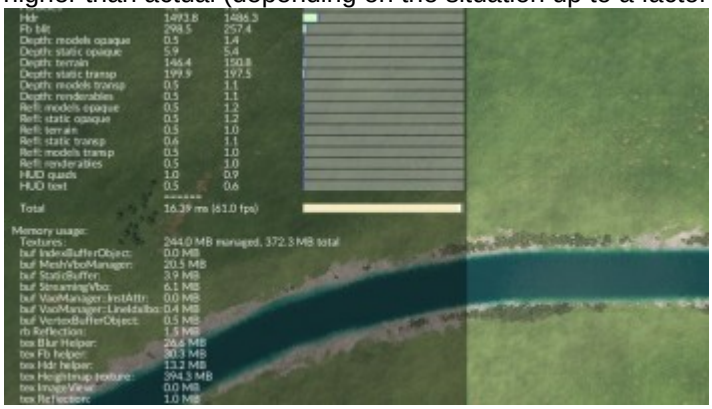
Many will likely already have enabled the FPS display of Steam. You can activate it in the Steam settings, I recommend it in high contrast at the top left. It updates itself every second, which can sometimes be too slow. But generally matches the actual FPS. With Freezes, however, it stops or sometimes disappears.

There are other external programs such as FRAPS, sometimes with more settings or even recording functions.

3 Debug Window (Rendering)

With the last update in March, new useful debugging tools were added, including rendering statistics (AltGr+I). Here the calculation times of the individual graphic elements (models, terrain, HDR, reflection...) are displayed, including the total time per frame with the resulting rate per second. Anyone who has already looked at this has probably already noticed that these values do not match the real FPS. This is because it is only about the calculation time of the rendering engine, script calculations and other things are on top (see below), therefore this value for the FPS is always higher than actual (depending on the situation up to a factor of 2).

If someone has more information here, please add.



Nevertheless, this window is great for examining the effects of graphic settings (grass, shadows, geometry) under different environments (city, train station, zoomed in or out, map size).

4 FPS Monitor

Now, if you want to determine systematic differences in performance, you have to rely on manual reading and estimating, which is of course not too precise, as the subjective feeling is often on top. The second point is that for meaningful comparisons, the FPS must always be recorded under exactly the same conditions (see influencing factors). These problems are solved with the FPS Monitor.

The use is explained in the mod entry: FPS Monitor

5 Script vs Graphic Performance

The calculations can be roughly divided into these two areas. In the game, graphics and script functions with calculations are relatively independent of each other and run in their own threads (at least in Lua, probably also in the C++ part). The FPS depend on both factors, it makes sense to differentiate between them.

- The **graphics calculations** run continuously and are independent of the game speed. For each frame, the current view must be rendered and the GUI must be displayed (see rendering debug window). The current position and the number of models to be displayed influence the calculation time. In principle, therefore it doesn't matter how many vehicles you have and how many residents the cities have.
- The **script calculations** contain all the simulations for vehicles, the decisions of all people, path algorithm of the lines and a lot of background calculations. The game speed plays an important role here: in pause mode, there is actually no need to calculate anything, with multiple game speed, more

https://www.transportfever.net/lexicon/entry/309-fps-display-measurement-analysis/

has to be simulated in the same time. In contrast, the current position does not matter because all objects on the map must always be simulated.

You can isolate these two components relatively well (not entirely). Depending on the savegame, one part may be more relevant. There are almost no script calculations on an empty map, but if you have a lot of vehicles and people (keyword Lategame), this will be the bigger problem than the graphics performance. On the other hand, if you have detailed, densely built-up cities, a lot of models have to be represented in these areas, so that the rendering has a lot to do here.

- If you want to test the **graphics performance**, it is best to pause and look for a suitable view. Ideally, this should always be the same one for comparisons, for which you can use the tools of the FPS Monitor or the functions of the camera tool.
- The best way to measure **script performance** is to minimize the graphics settings, go to the edge of the map and look into nirvana, so that the rendering has almost nothing to do. Set game speed to normal.

It is of course not forbidden to measure both at the same time. You can also let the game run in a suitable position, another scenario would be a cab ride. But you have to consider that in that case, the objects in the environment change, which can change the calculation time, which means that the FPS fluctuate. There is also a dependency on the starting point.

6 Influencing Factors

As mentioned before, there are a number of variable factors that affect FPS and should be kept in mind. When testing, however, it is important that it always run under the same environmental conditions, otherwise comparisons are senseless. Especially if you use the recording function of the FPS Monitor, you should note the following things.

6.1 External

How much computing power is available for Transport Fever depends, of course, first of all on the hardware, but also on the operating system and background programs (CPU and RAM). I don't think that an open browser changes the results significantly, but of course you shouldn't have too much open programs and watch out for sudden or irregular program starts like virus scanners.

6.2 Transport Fever specific

- First start: The game has to "warm up" at the first start after some time, which is because files are cached. If you use the autostart function of FPS-Monitor, it is best to measure the same scenario twice at first.
- Autosave: It is best to switch off for longer recordings so that there is no drop.
- Foreground/Background: If you minimize the game, the FPS increase a little. It is therefore best to always keep it in the foreground and full screen when measuring. For the script performance, the recording could also be run in the background.
- ESC menu: If you are in the save/load/exit dialog, the GUI thread is stopped so that there are no new frames. After continuing, this is registered by a long drop.

6.3 View

As indicated in the last chapter, the current position and orientation have a significant impact on the graphic performance. The calculations change depending on what is nearby, whether you are close to the ground or in the air and how many people/vehicles are currently walking through the camera.

6.4 Interaction/GUI

As soon as you scroll on the map or click on things, you may experience stutters and FPS drops. There is also a "warm-up effect" here, since map areas or models/constructions have to be read from the hard disk once. Of course, any interaction other than moving the mouse should be avoided during a recording.

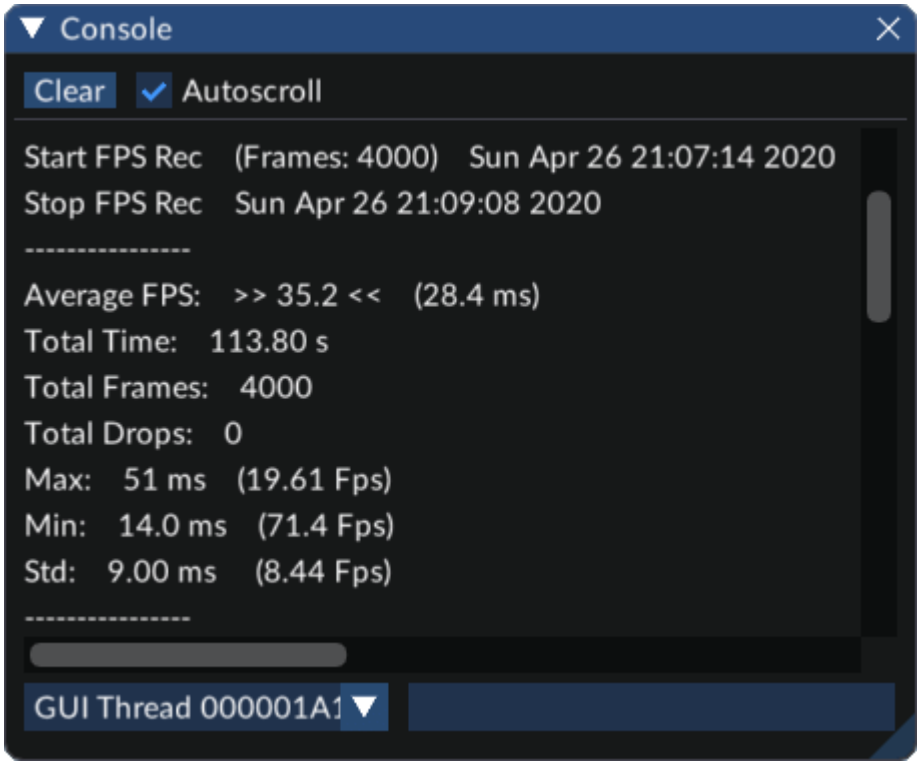
6.5 Console

The CommonAPI in-game console is very practical, but the performance is negatively affected if it contents too much. It is best to empty or hide the console before recording.

7 Analysis

Now that you've finally made a sensible measurement and the FPS Monitor has spit out a few values, the question arises as to what they mean. The average is undoubtedly the most interesting result. The fluctuation of the individual frame times is also important, which is why its standard deviation (std) is also calculated. The lower this value, the more stable the FPS and the higher the meaningfulness of the average value. In addition, the minimum and maximum values are interesting, as is the number of drops, of which as few as possible, better none, should be included.

For really reliable statements, measurements should also be repeated at different times and the recording length varied to exclude any influencing factors. The standard deviation should be less than about 20 ms and lower than the average. Of course, you should always add the test conditions to a claim with results.



[4] https://www.transportfever.net/lexicon/entry/309-fps-display-measurement-analysis/

8 Test ideas / Performance questions

This was quite a lot now, but you have all the information to systematically record FPS. Either for you personally to compare different things. But there are also countless open questions and discussions that could be examined more closely with the tool and the procedures presented here and I would be happy if you could contribute something with the FPS Monitor. Possibly, create new topics and let me know.!) Image not found or type unknown

One could also approach the many hardware discussions with it: A test savegame could be examined by different users with different hardware systems with the FPS Monitor. But make sure not to change the view after loading, so capture a suitable position when saving and go into pause mode. The parameters of the settings must always be added, of course.

Image not found or type unknown !?: A few questions as motivation!?: Image not found or type unknown

- What does the resolution change?
- What role do the graphics settings (textures, antialiasing, grass...) play?
- What influence does the terrain have, especially many terraforming?
- With how many residents, script and graphic performance is about the same?
- Do large forests have a significant effect?
- What influence do animals have?
- Do people consume less performance when using the lines than when driving a car?
- What influence do certain mods have?
- What does deleting the texture_cache bring?
- What do different settings of the graphics card affect?
- What does a frame rate limiter?
- Does "-USEALLAVAILABLECORES -high" make a difference?
- Is core separation (SMT) useful?
- From how many cores does the performance no longer increase?
- Does Task Priority help?
- What role does the paging file play?
- Which hardware properties do micro stutters depend on?
- Ratio between CPU vs GPU?