

# FPS Anzeige/Messung/Analyse

## Inhaltsverzeichnis

- [1 Einleitung](#)
- [2 Anzeige von externen Programmen](#)
- [3 Debug Fenster \(Rendering\)](#)
- [4 FPS Monitor](#)
- [5 Script vs Graphic Performance](#)
- [6 Einflussfaktoren](#)
  - [6.1 Externe](#)
  - [6.2 Transport Fever spezifisch](#)
  - [6.3 Ansicht](#)
  - [6.4 Interaktion/GUI](#)
  - [6.5 Konsole](#)
- [7 Analyse](#)
- [8 Testideen / Performance-Fragen](#)

## 1 Einleitung

Performance ist bei Transport Fever ein stets relevantes Thema und meistens ein Problem, das alle Leistungsklassen betrifft. Simulationsspiele wie TPF können im Allgemeinen nie genug Rechenleistung bekommen und Nutzer wollen auf immer größeren Karten mit immer mehr Fahrzeugen und Mods spielen. TPF hat dabei sicherlich ein paar spezielle Probleme, wobei ich davon ausgehe, dass Urban Games die Wichtigkeit dieses Themas bekannt ist und sich ja auch seit Train Fever bis zum letzten Performance-Patch schon einiges getan hat.

Es gibt bereits zahlreiche Diskussionen zu diesem Thema, in denen es um die Auswirkungen von Spieleinstellungen, Mods und einzelnen Hardwarekomponenten auf die Performance geht. Als Indikator für letztere werden eigentlich immer die **FPS** verwendet, wobei es neben dem durchschnittlichen Wert auch um die Stabilität bzw (Mikro-)Ruckler geht. Daher ist es wichtig, diese zuverlässig zu erfassen.

Da das manuelle Ablesen grundsätzlich mit Unsicherheiten behaftet ist, möchte ich zudem mein Mod-Tool **FPS-Monitor** vorstellen. Mit diesem wird das objektive und automatisierte Messen der Framerate unter gleichen Umgebungsbedingungen möglich. Dadurch können Auswirkungen detaillierter untersucht werden und Performance-Einflüsse präzise erfasst werden.

## 2 Anzeige von externen Programmen

Viele werden wahrscheinlich schon die FPS-Anzeige von Steam aktiviert haben. Die kann man in den Steam-Einstellungen aktivieren und am besten oben links mit hohem Kontrast anzeigen lassen. Sie aktualisiert sich jede Sekunde, was manchmal zu langsam sein kann. Stimmt aber im Allgemeinen mit den tatsächlichen FPS überein. Bei Freezes bleibt sie allerdings stehen oder verschwindet manchmal.

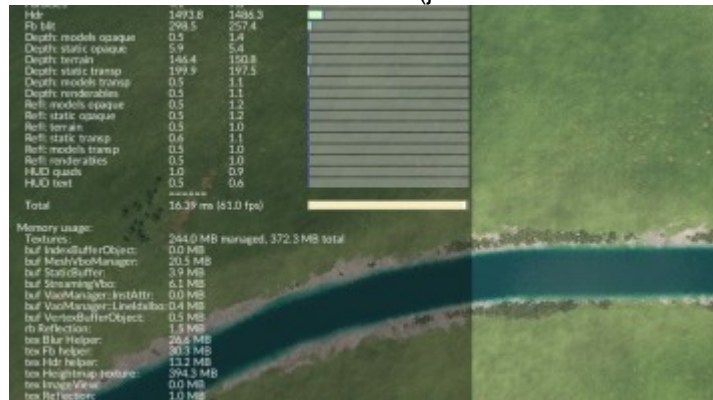
Auch andere externe Programme wie z.B. FRAPS ermöglichen die Anzeige, mit teilweise mehr Einstellungen oder sogar Recording Funktionen.

### 3 Debug Fenster (Rendering)

Beim letzten Update vom März wurden neue nützliche Debug-Tools hinzugefügt, unter anderem Statistiken zum Rendering (AltGr+I). Hier werden einem die Berechnungszeiten der einzelnen grafischen Elemente (Models, Terrain, HDR, Reflection...) angezeigt, inklusive der Gesamtzeit pro Frame mit der daraus resultierenden Rate pro Sekunde. Wer sich das schonmal angesehen hat, hat aber wahrscheinlich auch schon festgestellt, dass diese Werte nicht mit den realen übereinstimmen. Das liegt daran, dass es hier nur um die Berechnungszeit der Rendering Engine geht, Script-Berechnungen und anderes kommt noch oben drauf (siehe unten), daher ist hier der Wert für die FPS immer höher als tatsächlich (je nach Situation bis um Faktor 2).

Wenn jemand hier mehr Infos hat, gerne ergänzen.

Trotz dessen eignet sich dieses Fenster prima, um die Auswirkungen von Grafikeinstellungen (Gras, Schatten, Geometrie) unter verschiedenen Umgebungen (Stadt, Bahnhof, Ran- oder Rausgezoomt, Kartengröße) zu untersuchen.



### 4 FPS Monitor

Will man jetzt systematisch Performance-Unterschiede feststellen, ist man bisher allerdings auf händisches Ablesen und Abschätzen angewiesen, was natürlich nicht allzu genau ist, da oft das subjektive Gefühl dazu kommt. Der zweite Punkt ist, dass für sinnvolle Vergleiche die FPS immer unter den exakt gleichen Bedingungen erfasst werden müssen (siehe Einflussfaktoren). Diese Probleme werden mit dem FPS-Monitor gelöst.

Die Benutzung wird im Mod-Eintrag erklärt: [FPS Monitor](#)

### 5 Script vs Graphic Performance

Die Berechnungen lassen sich grob in diese zwei Bereiche einteilen. Im Spiel sind Grafik und Script Funktionen sowie Berechnungen relativ unabhängig voneinander und laufen in eigenen Threads (zumindest in Lua, wahrscheinlich auch im C++ Teil). Die FPS hängen von beiden Faktoren ab, daher ist es sinnvoll, jeweils zu unterscheiden.

- Die **Grafik Berechnungen** laufen kontinuierlich und sind unabhängig von der Spielgeschwindigkeit. Für jeden Frame muss die aktuelle Ansicht gerendert, sowie die GUI dargestellt werden (siehe Rendering Debug Fenster). Die aktuelle Position und die Anzahl der darzustellenden Modelle beeinflussen natürlich die Berechnungsdauer. Prinzipiell ist es hierfür aber egal, wieviele Fahrzeuge man besitzt und wieviele Einwohner die Städte haben.
- In den **Script Berechnungen** sind die ganzen Simulationen zu Fahrzeugen, die Entscheidungen aller Personen, Pfadfindung der Linien und jede Menge Hintergrundberechnungen enthalten. Hier spielt die Spielgeschwindigkeit eine wichtige Rolle: Im Pausenmodus muss eigentlich nichts berechnet werden, bei vielfacher Spielgeschwindigkeit muss mehr in der selben Zeit simuliert werden. Dagegen ist die aktuelle Position egal, weil immer alle Objekte auf der Karte simuliert werden müssen.

Man kann diese zwei Komponenten also relativ gut isolieren (wenn auch nicht ganz). Je nach Spielstand ist eventuell der eine Teil relevanter. Auf einer leeren Karte gibt es so gut wie keine Script-Berechnungen, hat man jedoch sehr viele Fahrzeuge und Personen (Stichwort Lategame), wird das das größere Problem sein als die Grafik Performance. Hat man dagegen detaillierte, dicht bebaute Städte, müssen in diesen Gebieten sehr viele Modelle dargestellt werden, sodass hier das Rendering viel zu tun hat.

- Will man die **Grafik Performance** testen, stellt man am besten auf Pause und sucht sich eine geeignete Ansicht. Diese sollte für Vergleiche idealerweise immer dieselbe sein, wofür man die Tools des FPS-Monitor oder die Funktionen des Kamera-Tools nutzen kann.
- Die **Script-Performance** misst man am besten, indem man die Grafikeinstellungen minimiert, an den Kartenrand geht und ins Nirvana schaut, sodass das Rendering fast nichts zu tun hat. Spielgeschwindigkeit auf normal setzen.

Es ist natürlich auch nicht verboten, beides gleichzeitig zu messen. Man kann das Spiel auch in einer geeigneten Position laufen lassen, ein weiteres Szenario wäre eine Mitfahrt. Man muss aber bedenken, dass sich dann die Objekte in der Umgebung ändern, wodurch sich die Berechnungszeit ändern kann, die FPS also schwanken. Außerdem ergibt sich eine Abhängigkeit vom Startzeitpunkt.

## 6 Einflussfaktoren

Wie schon erwähnt, gibt es eine Reihe von variablen Faktoren, die die FPS beeinflussen und man im Hinterkopf behalten sollte. Beim Testen ist es aber wichtig, dass sie immer unter den gleichen Umgebungsbedingungen ablaufen, nur so sind Vergleiche sinnvoll. Vor allem wenn ihr die Aufnahme-Funktion des FPS-Monitor nutzt, solltet ihr folgende Dinge beachten.

### 6.1 Externe

Wie viel Rechenleistung Transport Fever zur Verfügung steht, hängt natürlich zuerstmal von der Hardware ab, aber auch von Betriebssystem und Hintergrundprogrammen (CPU und RAM). Ich glaube nicht, dass ein offener Browser die Ergebnisse stark verändert, aber man sollte natürlich nicht unnötig viel offen haben und auf plötzliche oder unregelmäßige Programmstarts wie Virens Scanner achten.

### 6.2 Transport Fever spezifisch

- Erster Start: Das Spiel muss sich beim ersten Start nach Längerem erstmal "warmlaufen", was daran liegt, dass Dateien gecached werden. Wenn man die Autostart-Funktion des FPS-Monitor benutzt, deswegen am besten zunächst zweimal dasselbe Szenario messen.
- Autosave: Bei längeren Aufnahmen am besten abschalten, damit kein Drop enthalten ist.
- Vorder-/Hintergrund: Minimiert man das Spiel, erhöhen sich die FPS etwas. Daher beim Messen am besten immer im Vordergrund und Vollbild lassen. Für die Script-Performance könnte man die Aufnahme auch im Hintergrund laufen lassen.

- ESC-Menü: Befindet man sich im Spiel speichern/laden/beenden Dialog, wird der GUI-Thread angehalten, sodass es keine neuen Frames gibt. Nach Fortsetzen wird das durch einen langen Drop registriert.

### 6.3 Ansicht

Wie im letzten Kapitel angedeutet, hat die aktuelle Position und Ausrichtung erheblichen Einfluss auf die Grafik Performance. Je nachdem was sich gerade in der Nähe befindet, ob man nah am Boden oder in der Luft ist, wie viele Personen/Fahrzeuge gerade durchs Bild laufen, ändern sich die Berechnungen.

### 6.4 Interaktion/GUI

Sobald man auf der Karte scrollt oder Dinge anklickt, kann es zu Rucklern und FPS-Einbrüchen kommen. Auch hier gibt es ein "Warmlauffeffekt", da Kartenbereiche oder Modelle/Konstruktionen erstmalig von der Festplatte eingelesen werden müssen. Selbstverständlich sollte während einer Aufnahme jede Interaktion außer Maus bewegen vermieden werden.

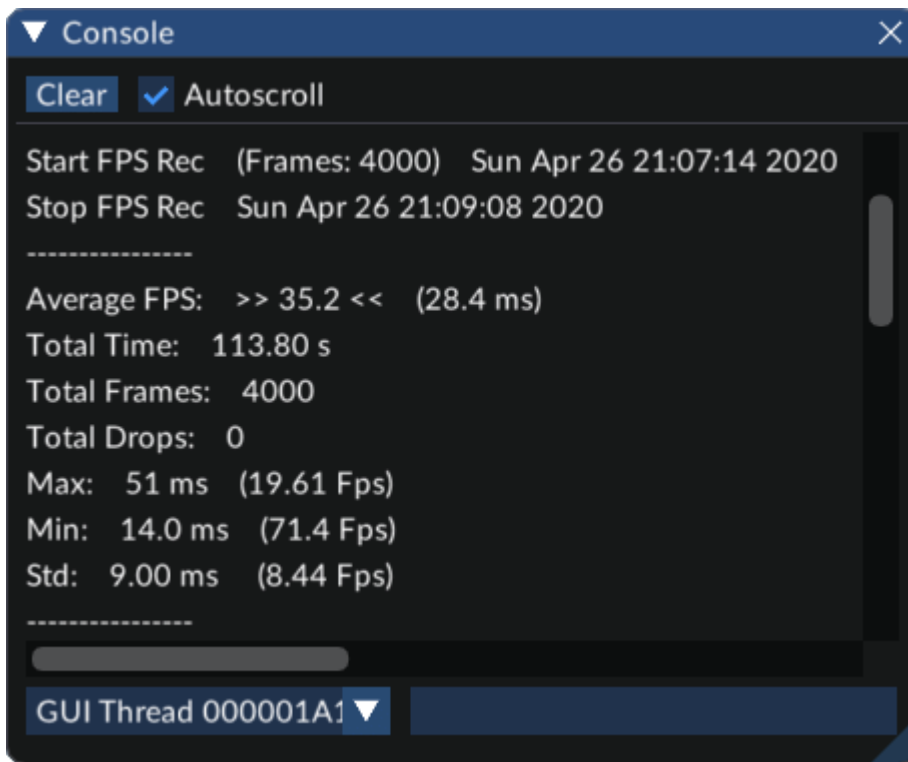
### 6.5 Konsole

Die Ingame-Konsole der CommonAPI ist zwar sehr praktisch, allerdings wird bei zuvielen Inhalten die Performance negativ beeinflusst. Vor dem Aufnehmen am besten leeren oder die Konsole ausblenden.

## 7 Analyse

Nachdem ihr nun endlich eine vernünftige Messung gemacht habt und der FPS-Monitor ein paar Werte ausgespuckt hat, stellt sich die Frage, was diese aussagen. Der Mittelwert ist zweifelsfrei das interessanteste Ergebnis. Aber auch die Schwankung der einzelnen Framezeiten ist von Bedeutung, weswegen zusätzlich dessen Standardabweichung (Std) berechnet wird. Je niedriger dieser Wert ist, desto stabiler sollten die FPS sein und desto höher ist damit die Aussagekraft des Mittelwerts. Zusätzlich sind Minimal- und Maximalwert interessant, außerdem die Anzahl der Drops, wovon möglichst wenige bis gar keine vorhanden sein sollten.

Für wirklich verlässliche Aussagen sollten Messungen außerdem zu verschiedenen Zeitpunkten wiederholt und die Aufnahmelänge variiert werden, um etwaige Einflussfaktoren auszuschließen. Die Standardabweichung sollte unter etwa 20 ms sein und niedriger als der Mittelwert. Selbstverständlich sollte man zu einer Behauptung mit Ergebnissen auch immer die Testbedingungen dazusagen.



## 8 Testideen / Performance-Fragen

So das war jetzt sehr viel auf einmal, aber ihr habt alle Informationen, um systematisch die FPS zu erfassen. Entweder für euch persönlich, um verschiedene Dinge zu vergleichen. Es gibt aber auch unzählige offene Fragen und Diskussionen, die man mit dem Tool und hier vorgestellten Vorgehen genauer untersuchen könnte und es würde mich freuen, wenn ihr mit dem FPS-Monitor etwas dazu beitragen könnt. Erstellt dazu gegebenenfalls neue Themen und sagt Bescheid!

Auch die unzähligen Hardware-Diskussionen könnte man damit angehen: Ein Test-Spielstand könnte von verschiedenen Nutzern mit unterschiedlicher Hardware mit dem FPS-Monitor untersucht werden. Achtet aber darauf, nach dem Laden die Ansicht nicht zu verändern, daher beim Speichern eine geeignete Position annehmen und in den Pausenmodus gehen. Die Parameter der Einstellungen müssen natürlich immer mit angegeben werden.

Als Motivation einige Fragestellungen:

- Was ändert die Auflösung?
- Welche Rolle spielen die Grafik-Einstellungen (Texturen, Antialiasing, Gras...)?
- Welchen Einfluss hat das Terrain, insbesondere viele Geländeänderungen?
- Ab wie vielen Einwohnern sind Script und Grafik-Performance etwa gleich?
- Haben große Wälder einen signifikanten Effekt?
- Welchen Einfluss haben Tiere?
- Verbrauchen Personen weniger Performance, wenn sie die Linien benutzen als wenn sie Auto fahren?
- Welchen Einfluss haben gewisse Mods?
- Was bringt Löschen des texture\_cache?

- Was bringen verschiedene Einstellungen der Grafikkarte?
- [Was bringt ein Frame Rate Limiter?](#)
- Macht "-USEALLAVAILABLECORES -high" einen Unterschied?
- [Was bringt Core Trennung \(SMT\)?](#)
- Ab wie vielen Kernen erhöht sich die Leistung nicht mehr?
- [Hilft Taskpriorität erhöhen?](#)
- [Welche Rolle spielt die Auslagerungsdatei?](#)
- Von welchen Hardwareeigenschaften sind Mikroruckler abhängig?
- [Verhältnis CPU vs GPU?](#)