Benchmarking Ethics and the ATI Radeon

October 24, 2001 - Kenn Hwang and Brandon Bell

Introduction

Stop the press!

Like the rest of the online press, we received our RADEON 7500 and 8500 review samples at the beginning of last week. At first, things proceeded smoothly; both boards were running great on our testing platform (in this case, a Pentium 4 since we've already examined the performance of Titanium boards on Athlon). Then we received word of HardOCP's results with a modified Quake 3 executable. It appeared as if the RADEON drivers was looking for Quake and optimizing for performance if it was found. By changing all references from "quake" to "quack" performance of the RADEON 8500 was slower. We were in the middle of testing the validity of the same program with an executable we'd made of our own that performed a similar function. By the time we'd completed testing with all three executables it was obvious that ATI had modified their drivers specifically for Quake 3. Therefore, rather than discussing the features and performance of the RADEON 7500 and 8500, we've decided to devote this entire article to examine exactly what's going on.

"Optimizing" or cheating?

Naturally whenever a company optimizes their product to perform well in one benchmark, suspicions are quickly raised. We've all heard the stories of companies who have cheated in synthetic benchmarks in the past. What makes ATI's latest move slightly different however is that they claim the changes they've made to their driver are "optimizations" for Quake 3.

Usually when a company optimizes the performance of their drivers for popular game engines or one game in particular this effort is applauded by the press and end users. However, driver optimizations have traditionally been accomplished via more efficient code. In the case of ATI's drivers, these optimizations are obtained by modifying the visual quality of textures; we'll discuss this in greater detail on subsequent pages. The question we'd like to address is the ethical implications of these drivers.

As we've discussed above, game-specific optimizations are certainly a good thing -- companies have been doing it for years. Even texture quality optimizations are nothing new. If you recall NVIDIA's 5.x driver release, texture compression was enabled by default. While this resulted in a significant performance gain, image quality was compromised. Unlike NVIDIA's experiement with texture settings in the 5.x Detonators, ATI's driver is different in the sense that its changes are undocumented and can't be toggled on or off by the end user. While this may be excusable to most gamers, it puts the development community in an awkward situation. How would a texture artist using Quake 3 check his work?

For hardware websites such as FS, test results obtained with ATI's drivers arguably shouldn't be compared to other cards. While ATI owners are forced to use the settings the driver imposes (and therefore its indicative of what an end user would experience), the settings used on the other cards aren't the same, giving ATI an unfair advantage. Quite simply it isn't an apples to apples comparison.

Whether or not this is a deceptive practice is for you to decide; right now we're going to explain what we believe ATI is doing in the drivers provided with the newer RADEON boards.

Naming Code

Quack Verification

When we first heard about the "quackifier" program over a week ago, our first impression was one of skepticism. All we knew was that this was a program that purported to change string values in an executable file. When run, it did create a new "quack3.exe" file, but what else did it do? It could have made more changes to the file than its claimed purpose, or even modify existing OpenGL drivers.

While we inquired into the quackifier source, we asked Andrew, our ace Gamers.com developer to create an application from scratch that does the same thing; namely, to search a single file for any string values containing "quake" or "Quake" and rename those to "quack". Within 5 minutes, we had a fully working version, shown below:

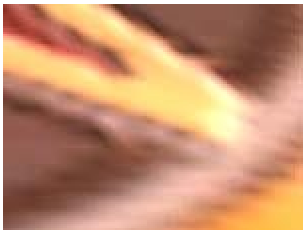
```
import java.io.*;
import java.util.*;

public class Test {

public static void main(String[] args) {

try {
//test string
String str = args[0];
String rep = args[1];
String inFile = args[2];
String outFile = args[3];

System.out.println( "Test String: " + str );
byte[] pattern = str.getBytes();
System.out.println( "Replace with: " + rep );
byte[] replace = rep.getBytes();

//read file into array
BufferedInputStream input = new BufferedInputStream
          (new FileInputStream(inFile));
ByteArrayOutputStream buffer = new ByteArrayOutputStream();
int b = input.read();
while (b != -1) {
          buffer.write(b);
          b = input.read();
}
byte [] file = buffer.toByteArray();
buffer.reset();
System.out.println(inFile + " size: " + file.length + " bytes.");

// search thru file & replace
byte [] test = new byte[pattern.length];
for (int i=0; i < file.length; i++) {
          if (file[i] == pattern[0] && (i + pattern.length) <= file.length ) {
                    System.arraycopy( file, i, test, 0, pattern.length);
                    if (Arrays.equals( pattern, test )) {
                              System.out.println("Match at: " + i);
                              buffer.write(replace, 0, replace.length);
                              i += (pattern.length - 1);
                              continue;
                    }
          }
          buffer.write(file[i]);
}

byte [] newFile = buffer.toByteArray();
buffer.close();
// write out new file if changed
if (! Arrays.equals(file, newFile) && ! outFile.equals("null")) {
          BufferedOutputStream output = new BufferedOutputStream
                    (new FileOutputStream(outFile));
          output.write(newFile, 0, newFile.length);
          output.close();
          System.out.println("Wrote new file: " + outFile);
} else {
          System.out.println("File not written");
}

} catch (Exception e) {
          e.printStackTrace();
}
}

public static void print(byte [] pattern) {
          for (int i=0; i < pattern.length; i++) {
                    System.out.print( pattern[i] + " ");
          }
          System.out.println();
          System.out.println();
}
}
```

Now that we have our quack, let's see what it does.

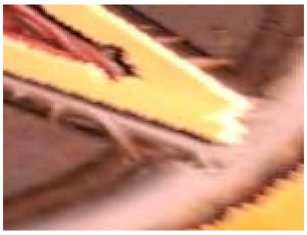
Image Quality

The Quack Changes

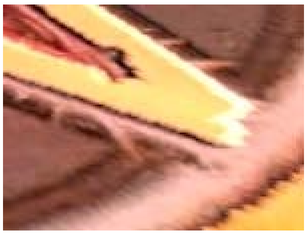
Before even Quack, it was noticeable that Quake 3 was changed. ATI has always rendered scenes slightly differently (in fact, the rendered outputs from different chipsets are never identical), but what we saw was very interesting. Below are images from the ATI Quake, ATI Quack, and an Nvidia GeForce 3 for comparison, in lossless PNG format.



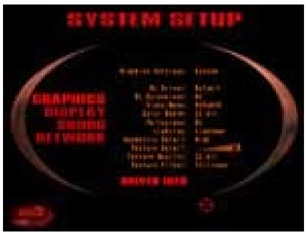
R8500 Quake3.exe



R8500 Quack3.exe

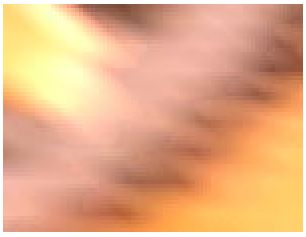


GF3 Quake3.exe

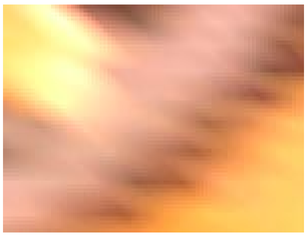


Best Quality Ever

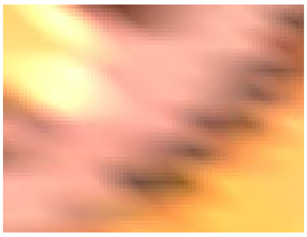
In the above screenshots, you can easily see that the large animated star texture is set to a low-quality texture in shot 1, even though we have explicity set Quake 3 to its highest-quality image setting. In fact, this particular texture is a pixel-perfect matchup for a low texture quality setting on an 8500 running the modified quack3, and a very close to the same setting on an Nvidia GeForce3, as the following 200% crops from the above images will show.



8500 Quake3.exe



8500 Quack3.exe



GF3 Quake3.exe

What does this do?

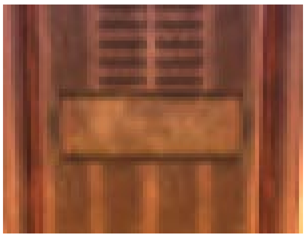
Quite simply, reducing the detail of the textures decreases the amount of memory required to store them, and thus amount of information needed to be sent down the pipeline. This reduction in required fill-rate should have a large impact at high resolutions, where large textures are frequently the bottleneck. You'll see this borne out in our benchmarks further on.

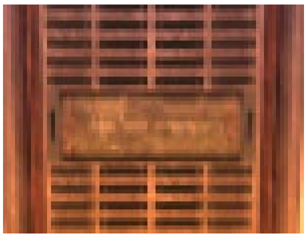
This alone could have been enough to account for the Radeon 8500's increased benchmark scores. But wouldn't it have been a bit too obvious to anyone who does more than simply run through the benchmarks?

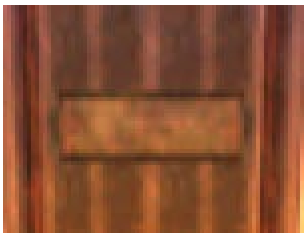
More Image Changes

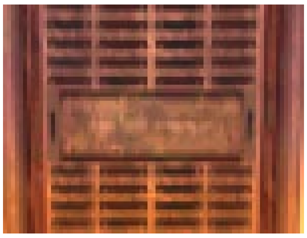
A bit confusing

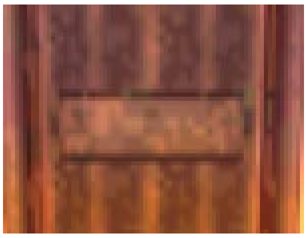
If ATI had simply disabled higher-quality texture settings and forced washed out, diluted images on us, the issue would be cut and dry - intentional disabling of high-quality settings to achieve faster framerates regardless of in-game settings. However, things are never so easy. A more detailed examination of the same scene yields some interesting results:


8500 Quake3.exe Best


8500 Quack3.exe Best


8500 Quack.exe Low

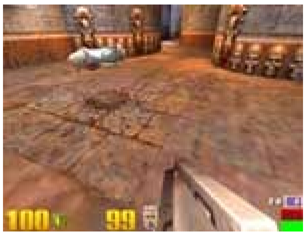

GF3 Best Quality


GF3 Low Quality

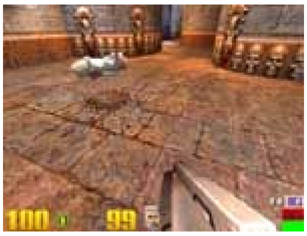
The first image is of course the "optimized" Quake3.exe from a stock Radeon 8500. Interestingly, what we see here is a mix between the high texture quality 2nd and 4th images, and the low texture quality of the 3rd and 5th. Notice that due to the downward angle of view in the original shot, the top of the crop is closer than the bottom, and is right at the boundary of the the bilinear filter. Apparently, the Quake-specific code in the ATI drivers will pull out more detail than the standard low texture-quality setting. The effect of this in-game is generally passable as medium-to-high texture quality, but with a 9-15% boost in framerate.

But wait, there's more!

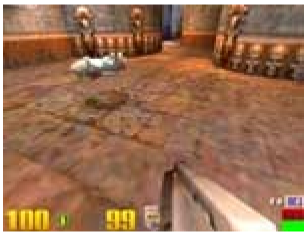
While it looks like texture quality and mipmap filtering are the principle culprits, there's even more: an odd-looking effect that very-easily gives away the non-standard drivers. Take a look at these shots:



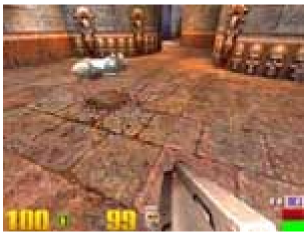
Something Strange?



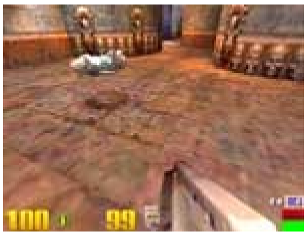
R8500 Best Quality



R8500 Lowest Quality

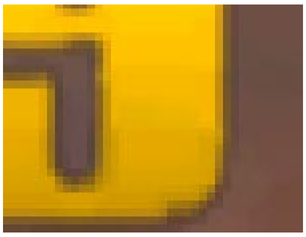


GF3 Best Quality

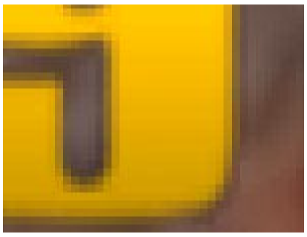


GF3 Lowest Quality

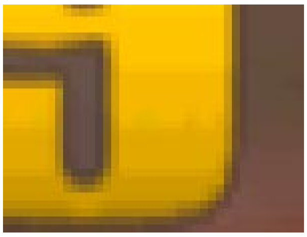
Also, a quick 400% crop of the health indicator shows even more, namely a few incongruous artifacts along smooth lines, and what appears to be 16-bit texture storage. This effect could also be a texture compression scheme, which would further reduce the bandwidth requirements at medium to high resolution.



8500 Quake3.exe
32-bit textures



8500 Quack3.exe
32-bit textures



8500 Quack3.exe
16-bit textures

Now that we have an idea of what ATI's "Performance Drivers" may be doing, let's run some benchmarks and see if our theories hold water!

System Setup

Test System Setup

Intel Pentium 4 2GHz

ABIT TH7-RAID

256MB PC800 RDRAM

ATI RADEON 7500
ATI RADEON 8500
Driver version 7.60
ASUS V8200 GeForce3 Pure
NVIDIA GeForce3 Ti 500
NVIDIA GeForce3 Ti 200
Driver version Detonator 21.85

30GB IBM Deskstar DTLA 307030 ATA/100 Hard Drive
AFREEY 12X DVD-ROM
Windows 98 SE
Windows XP Professional

DirectX 8.0a

Desktop resolution: 800x600x16

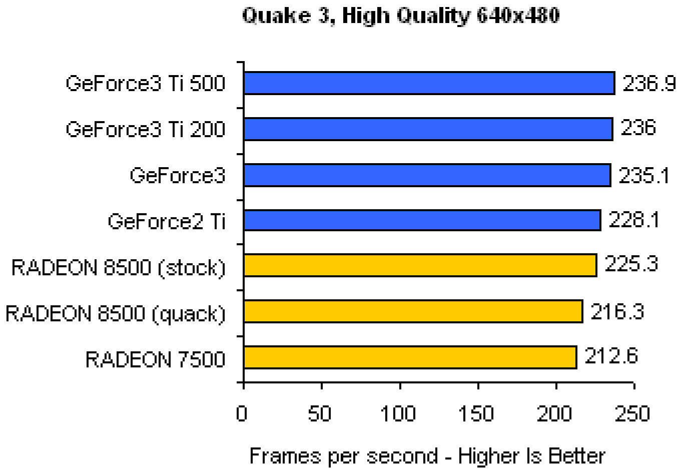
Benchmarks

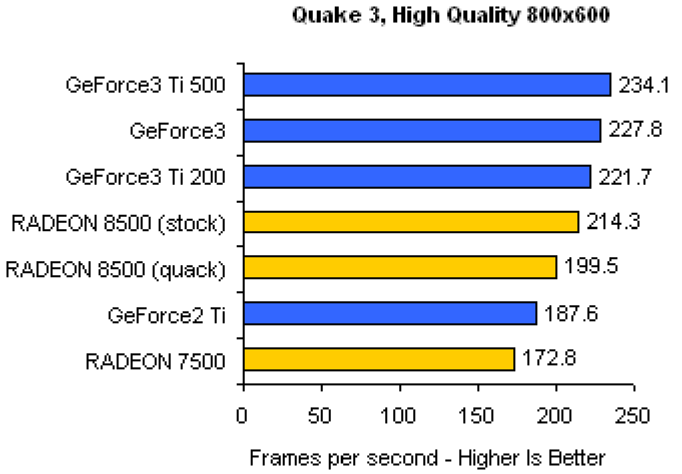
3DMark 2001

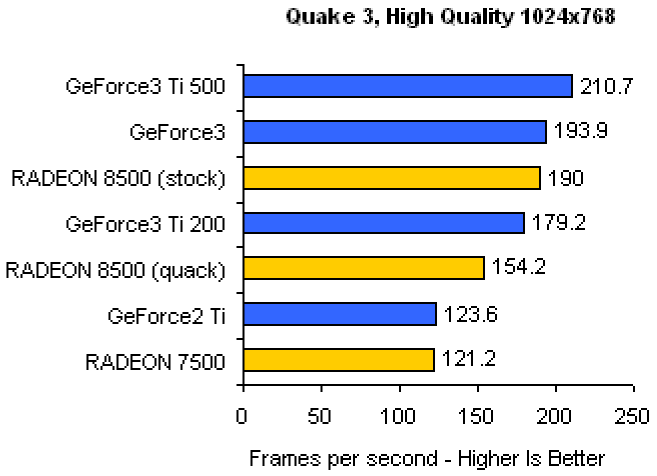
Quake 3 Retail - 640x480 High Quality
Quake 3 Retail - 800x600 High Quality
Quake 3 Retail - 1024x768 High Quality
Quake 3 Retail - 1280x960 High Quality
Quake 3 Retail - 1600x1200 High Quality

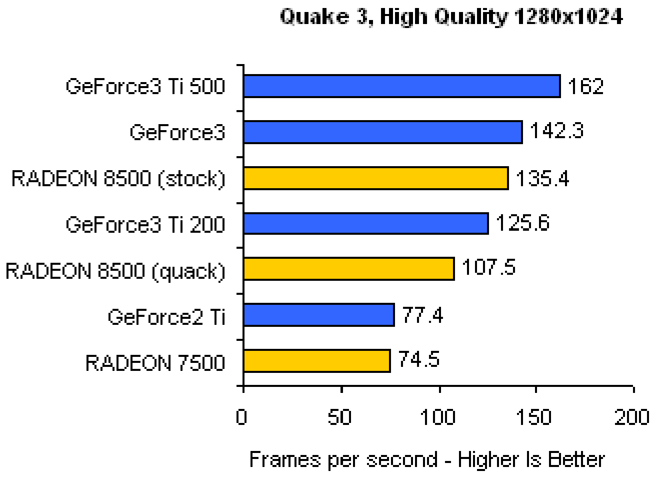
Quake3 Benchmarks

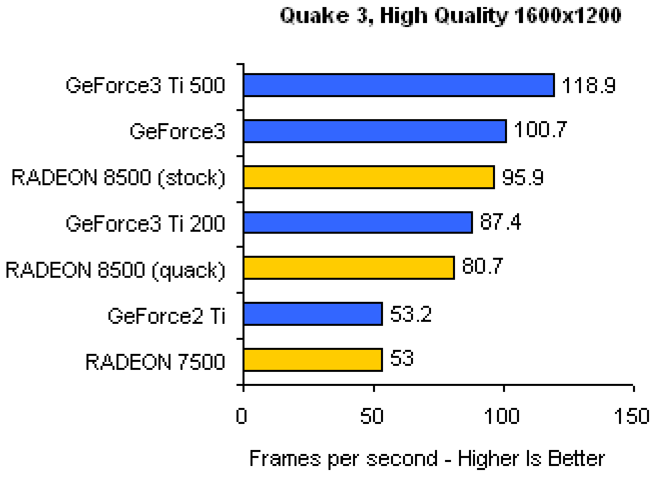
Quake III - High Quality











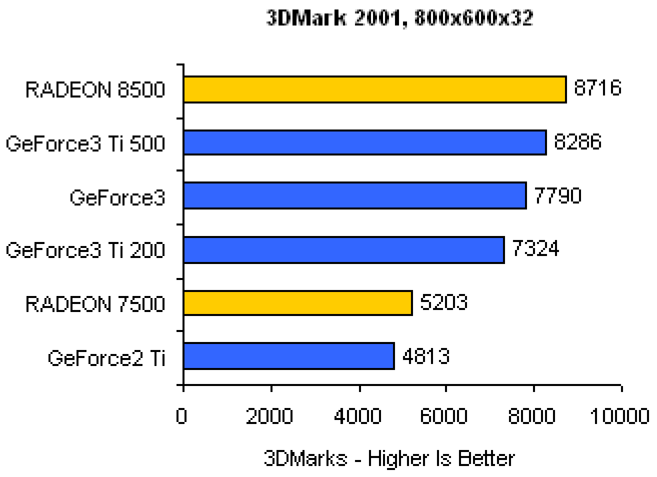
Analysis

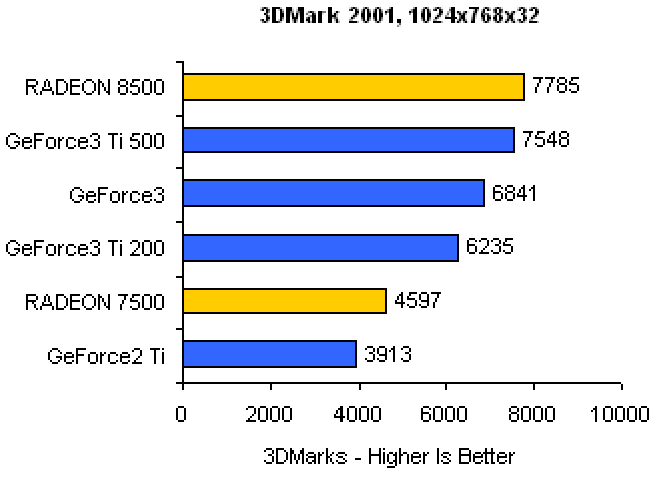
Even with stock drivers the RADEON 8500 isn't able to keep up with the GeForce3 and GeForce3 Ti 500 in high quality mode. We do see the performance increase it brings, roughly greater than 15% (although this figure does vary depending on the resolution). As you can see, without the driver modifications the RADEON 8500 finishes behind the GeForce3 Ti 200, thus saving a potentially embarassing situation for ATI.

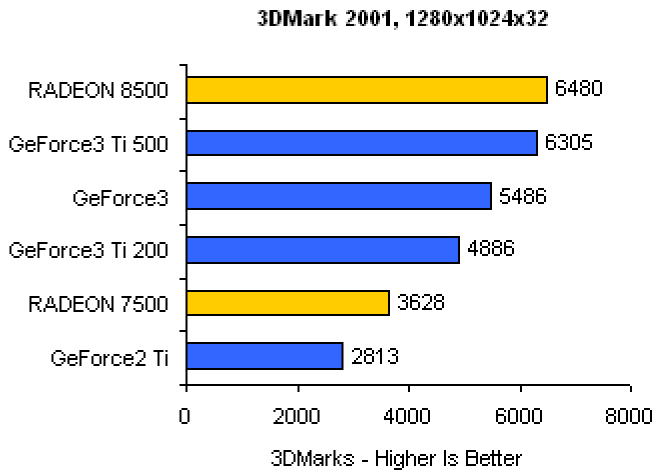
The GeForce2 Ti and RADEON 7500 compare very close to each other in performance. The RADEON 7500 results we've listed here are from the stock driver. When using our custom "quack" driver, the performance dropoff is typically less than 2%.

More Benchmarks

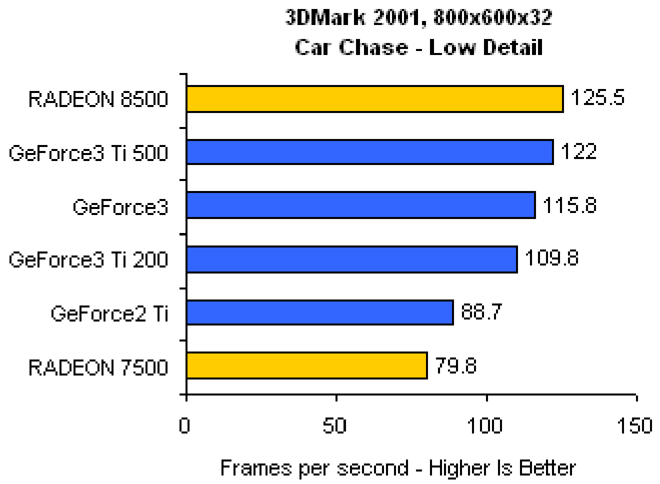
3DMark 2001 - DirectX 8.0

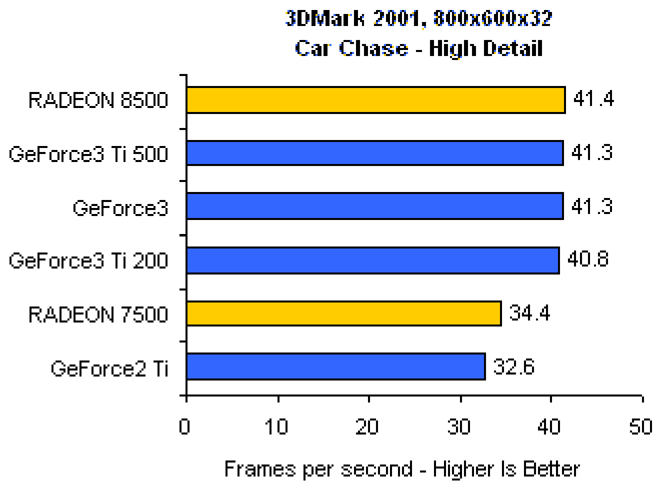




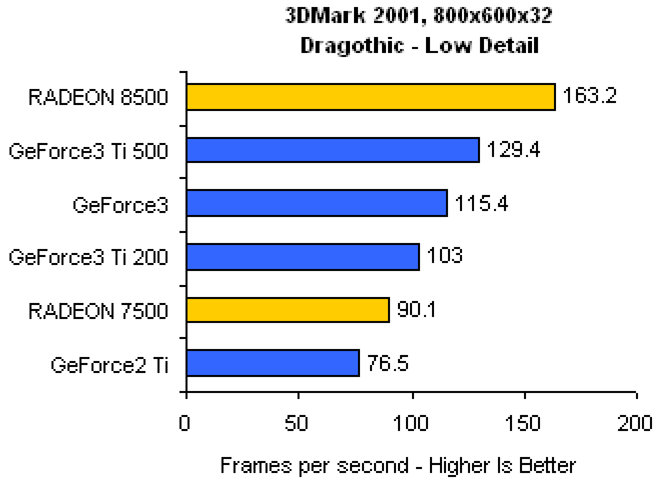


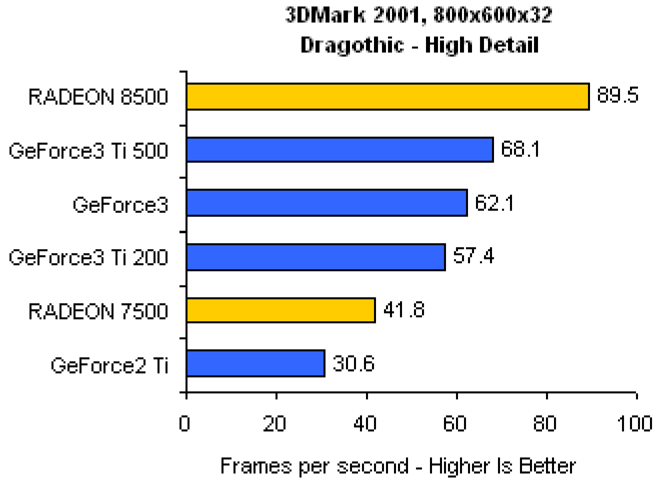
3DMark 2001 - Car Chase



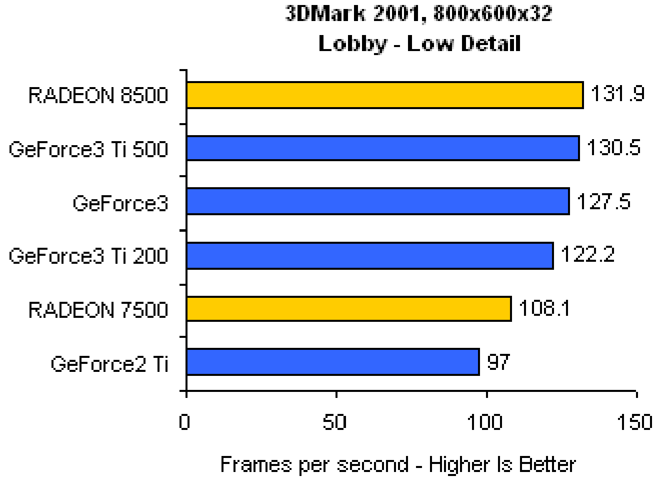


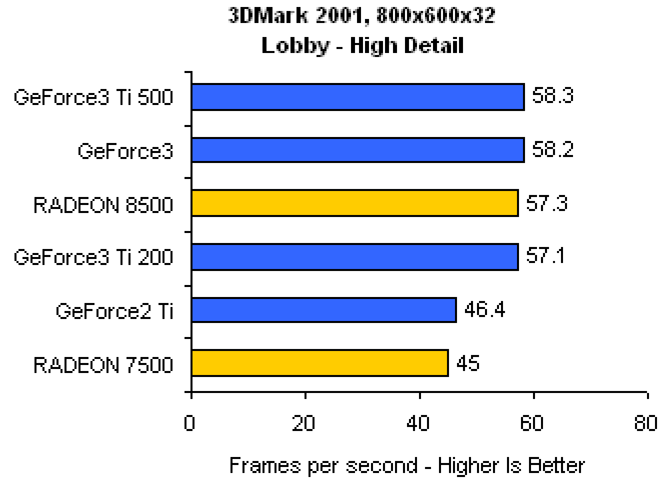
3DMark 2001 - Dragothic



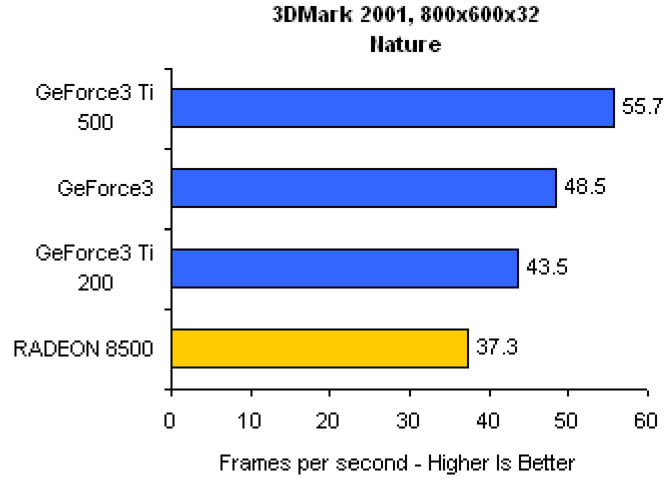


3DMark 2001 - Lobby





3DMark 2001 - Nature



Analysis

In 3DMark 2001, another popular 3D benchmark, the RADEON 8500 is able to outpace NVIDIA's GeForce3 Ti 500. This directly contrasts our Quake 3 results. As you can see, the RADEON 8500 really performs well in Madonion's "dragothic" demo, while the NVIDIA boards perform better than the 8500 in the "nature" test. Performance is neck and neck with the NVIDIA boards in the other tests.

The RADEON 7500 is able to outperform the GeForce2 Ti, thanks in large part to its superior performance in the same dragothic demo. The RADEON 7500 trades positions with the GeForce2 Ti in the other tests.

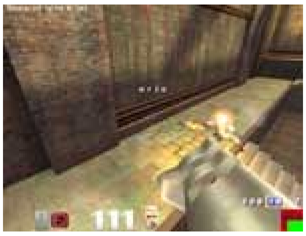
Conclusion

The Effect

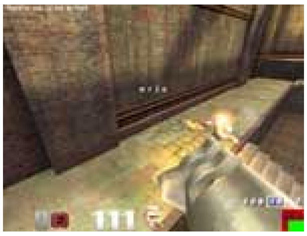
From what we've seen, it's clear that this is not an open and shut case. On one hand, ATI has made modifications to their drivers that make their hardware perform better on a popular game. There's nothing wrong with that, and even if the image quality is affected, most gamers running and gunning through servers would see the benefit of trading in a few textures for more frames per second. In fact, the fact that ATI can accomplish this while keeping visual quality generally high is commendable.

However, several facts point towards a more conspiratorial design. For one, as big as the Quake brand is, nowadays it's not even close to being the most heavily-played FPS. However, it IS a recognized, industry-standard benchmark; as such, it was a very convenient game for ATI to dedicate resources towards optimizing. Also, the modifications to the ATI drivers completely override the graphics settings available within Quake 3. This also conveniently leaves benchmarkers in the cold because they cannot effectively change game settings back to default; any comparisons with other drivers are video cards become essentially useless.

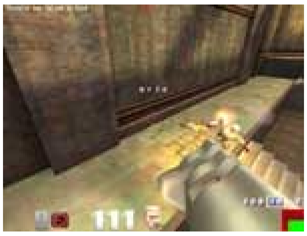
But most damning is the fact that ATI chose not to disclose its modifications to the public, or provide a way to toggle them on and off. Do such optimizations need to remain confidential? To what end does this serve? While the static screenshots seem obvious, it takes a careful eye to catch the drivers in the act in a dynamic game environment...and if benchmarkers miss it and erroneously attribute higher scores to the Radeon's performance, could we have counted on the manufacturer to set the record straight?



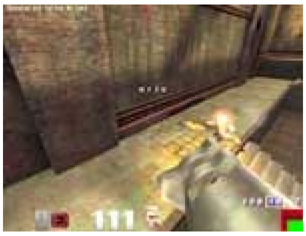
R8500 Quake3.exe
Best Quality



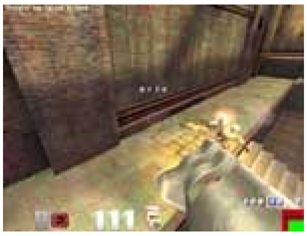
R8500 Quack3.exe
Best Quality



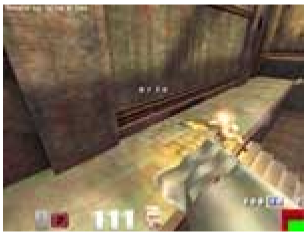
R8500 Quack3.exe
Low Quality



GF3 Med Quality



GF3 Best Quality



GF3 Low Quality

Conclusion

To some of us, it seems like the evidence points towards intentionally deceptive code designed not only to inflate benchmark scores, but also to keep anyone from finding out. To others, this is nothing more than an overreation to a perfectly legitimate game optimization. In our eyes, anyone who vehemently peddles either of these explanations is either naive or pushing an agenda of their own. Particularly true in this case, the truth lies somewhere between the two extremes. We have contacted ATI as well as Nvidia, and hope to bring you their responses soon. And while the sparks will fly, ultimately it is the consumer who will determine what constitutes acceptable behavior.