



*N*VIDIA™

Contentment

Richard Huddy, NVIDIA Corporation

RichardH@nvidia.com

Contentment?

- **Who are we trying to make happy?**
 - **IHVs**
 - **Publishers**
 - **Consumers**
 - **Our employers**
 - **Ourselves**



Contentment?

- **How can we make them happy?**
 - **IHVs**
 - **Use the hardware**
 - **Publishers**
 - **Deliver on time and to budget**
 - **Consumers**
 - **Give them a jaw-dropping experience**
 - **Our employers**
 - **Meet the schedule and budget**
 - **Ourselves**
 - **Be productive, be smart, be focussed**



Satisfying the IHVs

- **Take full advantage of the current hardware**
 - no use of multi-pass when multi-texture will do it
 - Hardware TnL as a default
 - Shaders in all future games
 - Don't supply fallbacks right back to Riva128
- **Deliver product launch material**
 - Anything that really shows off the new hardware
- **Supply an honest, transparent benchmark mode**
 - “Apples to apples” is the key phrase



Satisfying the publisher

- **They worry about two things above all else:**
 - **Cost**
 - **Schedule**
- **That's one reason it helps to work with the IHV's:**
 - **IHV's put together demo material you can cut and paste straight into the app**
 - **IHV's will help you to deliver features as reduced cost to you**



Satisfying the consumer

- **High frame rates**
 - Default to the main res of the machine (1024x768 is commonest)
- **High standard of QA**
 - IHVs will help you out here too – but we need to get a copy of the game before it's released
- **Use all the features they spent their money on**
 - Or it's just expensive sand
 - This is an important part of the buyer's psychology



Satisfying your employers I

- **Be productive**
 - That means taking advantage of all the resources available to you. (MS, IHV sites, Intel/AMD, source control, sample game code, gamasutra etc)
- **Be on top of the technology**
 - Understand where it's going so that you can plan ahead... (Hopefully – that's why you're here)
- **Be on schedule**
 - Understand the dynamics of this business and know what costs are introduced when you make certain decisions. [Mention Scalability Here!]



Satisfying your employers II

- **Help to coordinate the team**
 - **Art**
 - **If the art dictates the code then you're in trouble...**
 - **And art scales well downwards not upwards (i.e. it's easy to reduce complexity, it's hard to increase it)**
 - **Sound**
 - **It's trivial to take advantage of the best sound hardware if you plan the assets in advance**
 - **Multi platform considerations**
 - **Make sure that each team understands how their decisions may impact other platforms**
 - **But don't make a top end PC look like a PS2!**

Satisfying yourselves

- **Technical people love playing with toys**
 - Like cool new hardware
- **There are terrific learning opportunities here**
 - DirectX gets revised pretty often...
- **Let's not kill ourselves with mad schedules**
 - Distinguish between what's doable and what's not
- **Make the right decisions early enough to get the benefits**
 - It's really hard to optimise a game at beta...



The simple steps to contentment

- **Some hardware features are really easy to get at:**
 - 32 bit color,
 - Hardware TnL,
 - High precision Z buffer,
 - Antialiasing:
 - Just ask for it when you start things up
 - Chips which offer it are usually good at it
 - Multiple screen resolutions
 - I recommend that you default to 1024x768 unless the user is using a lower resolution desktop
 - High quality texturing
 - Choose anisotropic filtering or DXTC where it helps

The next level of contentment

- **The harder steps towards contentment all carry long term value for you as an engineer:**
 - **Cube-maps**
 - **Pixel shaders**
 - **Vertex Shaders**
 - **Bump mapping**
 - **All these things are very much a part of your future**
- **Things like fixed function matrix palette skinning or tweening just don't fit...**
 - **Q: "How do I bump-map that skinned thing?"**
 - **A: "Errrm. You can't." ☹️**

Learn to use a profiler I

- **I use VTune but there are others too...**
 - **Understand where your CPU cycles are going**
 - **Spend them wisely (i.e. be mean with them)**
- **If you don't currently use one then a profiler should change your life**
- **Understand it's limitations too...**
 - **It only describes one of the two main components**
 - **The GPU is left out of this description**
 - **But there are clues in the driver section...**
 - **And many IHVs have other tools which add valuable extra information (under NDA usually)**



Learn to use a profiler II

- **Profilers are probably the least well used commonly available resource**
 - **They're not wildly expensive**
 - **IHV specific versions are available**
 - **Like "NVTune Analyser" from developer.nvidia.com**
 - **And at least one other IHV has similar tools**
 - **"I wish I had the time to learn how to use a profiler"**
 - **I'd guess that it would cost you 2-3 days to learn**
 - **And many teams simply don't have this skill...**

Text of an email I got this week

- **‘Richard with your help and Vtune, one of my colleagues and I were able to increase the performance of our driver from 350% of competitor's time to 116% of competitor's time. Thanks for your help.’**
- **That’s over 3 times as fast as before**
- **And it’s not DirectX related – it’s just expertise related**
- **And it’s not so very unusual...**

Some common mistakes

- **“I tried that optimization and it didn’t go any faster, so I took it out”**
 - You may just be swapping bottlenecks
 - But removing the last one is the big win
- **“Englishmen are not sexy”**
 - Why would anyone believe this?
- **“I only use DrawPrimitiveUP once per frame so how bad can that be?”**
 - It can be wretched
- **“I need to sort my polys when using alpha test”**
- **“Passing the index set slows DIP() down”**



Big Mistakes

- **Not understanding how performance varies with the changes that you make**
 - You cannot measure performance by measuring the parts and summing them
- **This is inherently surprising to people who think of it as one machine – but remember that you would only be measuring the CPU side**
 - “I took out each Lock() in turn and it made no difference... so I put it back”, or,
 - “I’ve measured how long it takes to do that – that’s not the problem”

Big Mistakes

- **Limiting yourself thru the art**
 - **Once it's done this is very hard to undo...**
 - **If your art forces you to draw your scene as quads – then you will never get great performance**
 - **So the “performance problem” requires you to anticipate this and educate the artists “up front”**
 - **fail to do this and you're dead...**
 - **The “quality problem” is avoidable because your artists probably produce high quality assets**
 - **That's separate layers for all the effects**
 - **If they give you pre-lit textures then things look bad**



Big Mistakes

- **Mis-managed VBs**
 - **Understand the flow chart for VB usage**
 - **Take advantage of the driver-specific code underneath your app which knows what the hardware needs**
 - **Although common this one is not too hard to fix...**
 - **And... the tools out there help with the diagnosis**



Big Mistakes

- **Thinking that hardware TnL has the same perf profile as software TnL**
 - **It doesn't**
 - **Hardware TnL is even more sensitive to poor batching**
 - **And Hardware TnL is more important to you...**
 - **That's just a fact of the installed base**
 - **NVIDIA have sold more H/W TnL parts than S/W TnL**
 - **Plus the recent installed base is where you sell games**
 - **And it only gets to be more important as time goes by**



What's a good target for a game?

- **GeForce 2 / Radeon style graphics:**
 - 80K polys per frame
 - < 500 DIP calls
 - < 500 VB changes
 - < 200 texture changes
 - < 200 State change sets (SRT = SetRenderTarget)
 - “Few” SRT calls (that's single digits...)
 - Runs at refresh rate of 60Hz or better
 - That's less than 5 million polys per second
- **A GeForce4 Ti should happily do 4X this much...**
 - that's 250K polys per frame and no other changes to these targets



What's a good target for the future?

- **DX9 style mainstream graphics:**
 - **> 0.5 million polys per frame**
 - **< 500 DIP calls**
 - **< 500 VB changes**
 - **< 200 texture changes**
 - **< 200 State change sets**
 - **“Few” SRT calls (that’s single digits...)**
 - **1 pass per poly is typical, but 2 is sometimes smart**
 - **Runs at refresh rate of 80Hz or better**
 - **That’s better than 40 million polys per second**
 - **And nothing goes through the fixed function pipes**



Summary

- **PC graphics are about to exceed the capabilities of the Xbox in terms of both speed and quality**
- **So setting these targets high is essential if you're going to get the best out of the machine**
- **It's not hard to get the PC graphics pipeline right**
 - **Trouble is... it's also easy to get it wrong**
 - **And some mistakes are very hard to undo**
 - **So plan your strategy ahead of time**

Questions?



Richard Huddy
NVIDIA Corporation
RichardH@nvidia.com



NVIDIA