

---

## Diablo's Internal Speed

(Thursday, 02 December 2004) - Written by Jarulf

Yet another informative article by Jarulf captured from a forum post and preserved here with his permission. This explanation of the internal speed division of the game is useful as a reference to those new to the concept of Diablo and Diablo II's internal "frames per second" versus the more traditional definition of the term.

Bolty's Note: Yet another informative article by Jarulf captured from a forum post and preserved here with his permission. This explanation of the internal speed division of the game is useful as a reference to those new to the concept of Diablo and Diablo II's internal "frames per second" versus the more traditional definition of the term.

The reason people (partly erroneously) refer to the internal update rate as "frames per second" is probably due to the fact that people used those words in D1 when we referred to the update speed of the game. Now, D1 did not display any numbers on how often the game updated the screen, so there was no real confusion. Then came D2, which displays how often the game manages to update the screen and uses the "fps" word. This is probably more or less correct, since that is what one normally says in typical FPS and other 3D games when one refers to how often the game updates the screen.

As you say, the update of the screen has NOTHING to do with the internal update of the game; at least, not in D2 (although it can be tightly linked in other games). Also, don't forget that neither of those two numbers has anything to do with how often the monitor updates its picture; my monitor, for example, has at least 85Hz (85 times per second) for most resolutions, including those that D2 uses.

So what do all these things mean? Well, first, we have to realize that D2 is not a 3D game. All animations, monsters, spells, whatever, are basically shown as pre-rendered pictures (one could call them sprites). There is a set number of pictures for each action or animation, all fully drawn and stored on your CD. Take a Barbarian swinging an axe - the game may then have, say, 18 different pictures for this - which when put together one after another, show a Barbarian swinging his axe. How long will it take to show those pictures? Well, a fast computer would be able to shuffle all those pictures to the graphics board MUCH faster than your monitor would be able to update the screen. here is where the internal update rate of the game comes in. Internally, the game updates itself once every 1/25th of a second. It then recalculates the position of things, what actions to take, if a particular missile hit something, and so on. For each such update, it makes sure that everything (well, most things) in the game is updated; it basically goes through each monster, each player, each missile, and so on, and updates them according to the current action or starts a new action if the last one has ended.

The game can NOT calculate the state in between such updates. This means that a missile will basically move in steps each "tick." How far depends on the speed of it, of course. The game will never be in a state where the missile is located in between two such positions, since the game only updates itself 25 times per second.

Now, in addition to updating itself, the game has to show the current state on the monitor. It will thus build a screen to display based on the current state of the game. It cannot put something (for example our missile mentioned above) on the screen in between two positions calculated in subsequent ticks, since the missile will never be there. How often should the game build such a screen? Well, it really makes no sense to build it more often than it updates itself internally, since if it would, it would be identical to the one before. For example, assume it updated the screen 50 times per second. Then only every other update of the screen would have a change, since for the other there would not have been any update of the game state. This building and updating of the screen based on the game state is the "true" fps, that is, the value shown on screen as the fps (sure, there might be slight differences or other things that cause it not to be exactly as I describe, but the basics like this simplify explanation).

Also note that the monitor will still fetch whatever screen was last built for display at whatever rate it was set to, for example 85Hz as my monitor typically is. This has nothing really to do with the game or program or anything; it has to do with the speed at which the "beam" in the monitor paints the picture. Of course, should the game or computer update its screen 200 times per second, I would never notice that on my monitor, since it only fetches the data for the screen 85 times per second. However, this has really nothing to do with what we discuss here.

If you have a slow computer, it would NOT be fast enough to both update the game (the state of players, monsters, missiles and so on) AND generate a screen for it 25 times per second. There is simply too much to do for it to handle it all in 1/25th of a second. The time to update the game varies with the amount of things going on in the game, of course. If there are 100 active monsters, it takes slightly longer to update the game each tick than if there is only one active monster (that is why the game deactivates areas where there are no players - no need to waste computing power on things that no one would really notice). In addition, putting out 10 firewalls also takes some time, since each firewall is made up of many, many flames, each one updated separately. Also, the time it takes to build a screen from the state of the world depends on how much is going on. Here, we only have to bother with things that actually are on the screen. Things one screen away, of course, need to be updated for what they do and so on, but we don't have to bother with them as far as building a screen goes. Here is where firewalls come in and slow down everything (for example) - the

---

game will build the screen for each frame; if they are transparent, we'll also need to calculate a whole bunch of things for that. We'll have to handle all the light that it causes, and so on. Sure, modern 3D cards can do that, but since the game is not fully 3D, the processor still has to work quite a bit (it is not exactly true that it is not 3D, but you get the idea).

So what happens when the computer can't cope? Well, in single player, it is easy. The game can simply slow down, only updating itself as fast as it can while still creating the screen once for each internal update (tick). But in multiplayer, this doesn't work. The server is working at a constant 25 updates per second and you simply MUST keep up with that or everything gets completely out of synch. We thus have to cut down on the updates of the screen while still making sure we update the state of things at 25 times per second. Depending on how the game code is written, the client may actually not have to update everything in the world every tick; as long as the server can update the client about things when needed, the client could, for example, skip updating off-screen monsters. The server still handles that. In addition, the server does not have to bother about making a screen out of it at all - only the client needs to do that. Still, the client needs to keep up with the server 25 times per second, and if it can't, it has to skip updating the screen sometimes. How many times? I would guess this is the number shown on the screen; you will normally not see any frames skipped if your fps is 25 or above, as there is no need to skip any screen updates then - there have not been any updates to the game state. It seems that the values shown on the screen for fps and skipped frames are somewhat average values though, so you would probably at times see skipped frames even if the fps is currently showing above 25 fps.

OK, now assume that we have this super fast computer; not only can it update the game state 25 times per second and create a screen for each such update, it could manage to do that 100 times per second without a sweat. Why doesn't it do that? Well, here comes the fact that all things shown on the screen are pre-rendered animations. Now, assume the Barbarian is swinging his axe again. Such a swing should take, say, 1 second. With an update of 25 times per second, we need to have at least 25 pre-rendered pictures showing various stages in the swing to put on the screen. That is exactly what the game has, and they are stored in the various mpq files on your CD (or hard disc). Now, assume this super computer wanted to update everything (game and screen) 100 times per second - one would have to adjust the calculations for the update of internal things. For example, should a monster regenerate less life per update than if we update at 25 times per second? Should a missile need to travel a shorter distance? And so on. Not really a problem in itself, BUT what about the pre-rendered pictures for the animation? We would actually need 100 pictures!!! In between each picture needed before, we would need an additional 3 pictures showing the character in a position of the swing slightly different than the previous ones we had. That would, of course, require 4 times as much disc space, or another 3 CDs of space (and tons more memory to store it all in). Then comes the next person with the computer that handles 200 updates per second, and we need even more, and so on. Blizzard settled for 25 updates per second. This means that EVEN if your computer could update everything more often, it would still not have more animation pictures to show, meaning that it would have to show the same pre-rendered picture more than once. With that in mind, there is no need to calculate the state the character is in with higher precision than 25 times per second either, and there is no need to build a new screen more often either. Anything above 25 is just a repetition of the same.

Sure, one could update the position of a missile with more precision even if one does not update the picture of it more often. There are other things that one could also update more often, but it would probably be more work with very little gain. People with problems with slow frame rates would not notice it, and besides, 25 updates per second is really quite enough for smooth movement of things (which is one of the main things one actually COULD update more often without worrying about not having more pictures to show it). Actually, the more something moves, the less one benefits from more frequent updates (with some limits). There are even games that purposefully add blur to fast moving things to make them look better in motion, thus fooling the human eye. The eye can happily fill in for the movement in between the position of, say, a missile whose position only changes every 1/25 of a second.

So, how come the game shows values for fps above 25? Well, I would say that they probably have the game actually update the screen as often as it can, despite such extra updates just being exact copies of the previous ones. It then shows how fast it potentially could update. This sort of gives you the idea of what margin your computer currently has. A value of 50 tells that even if there would be tons of more things going on on the screen, it could cope. No idea why they "capped" the value in single player, though.

Now, a typical "3D game" works differently. There, the monsters, players, and so on are not shown as pre-rendered pictures in a sequence; they are instead (simplified) made up of various surfaces and rules about how they should move relative to each other when, for example, a monster moves. The game then has textures to display on these surfaces; thus, for each screen, it actually builds the picture of the monster, player, or whatever the object is (this was pre-done in D2). This means that there are, theoretically, an unlimited number of pictures to show of something. The game can update the screen as often as it wants, and each update would be calculated when it is supposed to be shown (instead of pre-rendered). It then makes much more sense to update the state of the game once for each update of the screen. The update of the game would have a factor related to how often the update occurs that scales how much something is updated each "tick". That is, the distance a player moved each update depends on how often the game updates. How far the arm that swings the axe has moved would also depend on how often the update occurs, and so on. Here the computer would simply update as often as it could. Sure, not everything might need updates each "tick", but we discuss the principles here. Still, as soon as the game manages to update itself and the screen faster than your monitor does, it is a waste. Also, the human eye can not distinguish updates that happen too fast. Most people really don't have any

---

problem with an update of 60 times per second; it appears perfectly smooth and increased updates would not look any different. Some people CAN see the difference though, and would say 60 updates (of the monitor screen) per second is flickering; they can actually SEE the update. Still, above 85 updates per second, hardly anyone has any problem.