

This interview with Jonas was conducted for the blog of Microsoft Student Partner Rasmus Boserup (aka. Razumasu) in January 2008. The topic of the interview, as of the blog, is general game design.

Just to get the people who don't know TNM up to spec, can you tell us something about how the whole idea for TNM came up?

TNM was conceived in a fairly random and uncoordinated way. Lawrence, our project lead, was messing around with the SDK when he had an interesting idea for a small mission where he would feature the members of the PlanetDeusEx bulletin board as characters. Everybody wants to see themselves in a game of course, so a lot of people chimed in with offers to help out, and within a couple of weeks he had a team of about 10 people working on it. I joined because they needed a story for the mod (back then the community was full of people with mod ideas but no skills or teams to realize them, so to find a full mod team in need of a story was too good a chance to pass up). The logical choice was to go all the way: Not just use characters based on real people, but actually set the whole mod within an interpretation of this virtual environment. It grew from there.

I know that you have a special view on how the story and gameplay of a game should play out. I'm referring to your blog post about linearity in games, how does this affect TNM?

Well I enjoy the occasional linear shooter as much as the next guy, but I think the most profound and memorable gaming experiences I've had came from interesting choices I've been asked to make in games. Interactivity is arguably the primary distinction between games and movies, so the more open and responsive a game's story and gameplay is, the more suited it seems to the medium (though it may not necessarily make the game more fun, of course). Deus Ex was one of the first shooters that really understood this and acted on it, for example making your allies respond to how many people you kill or subdue. We've done our best to maintain a lot of the design ethos from the original game, building on it instead of changing it too much. That means our levels are generally pretty open so the player can make his or her own way through them. It also means we've done our very best to make characters respond realistically to how you play the game. Kill a lot of enemies and some people will reward you, others will scold you, just like in Deus Ex. Knock somebody unconscious to steal from him early on, and he'll attack you the next time he sees you.

We weren't content with just matching Deus Ex' interactivity though, so we've implemented more side missions than Deus Ex had, and we have 2 storylines that run parallel through most of the game, which you're forced to choose between in the beginning. In addition to these two storylines, we have two warring religious organizations the player can choose between and complete a few missions for, and our main storylines both branch a little near the end, so you end up with 9 possible endings to the main storyline. I don't think anybody actually expects this much branching and nonlinearity in the plot of a commercial game, but precisely because we have no financial pressure, we've been able to go a little crazy with the amount of options we afford the player. Of course it helps that we know our primary target audience is people who loved Deus Ex so much they're still replaying it and finding new details in the game. Besides, this may be the last time any of us have a chance to make a game like this, might as well grab that chance

and make the most of it!

It's no secret that TNM has been under construction for a very long time. Can you tell us something about why the game has been in production for so long, and what your thoughts are about the entire process? Being wiser now, what would you all have done differently?

Well being wiser now, the game would only have taken 2 years to create. By far the most time has been spent learning game development, figuring out what worked and what didn't, not to mention learning the Unreal Engine 1.5. I guesstimate that 95% of the content we created during the first year has been scrapped including the storyline that was rewritten thrice, and a similar percentage may well be over 50% for the second year as well. For example, *all* our characters were recreated after the second year, when our character artist came back after a hiatus and realized how much his skills had improved. Most of our new weapons have also been remodelled at least once.

The second reason we've taken 6 years so far is because people routinely drop off the face of the Internet for months at a time, and by now we only have one 3D modeller, one texture artist, one dedicated level designer, etc., so any time one of them stops actively contributing, that whole part of our development just stops progressing. And finally, *feature creep*. The whole mod is one big pile of feature creep. Until a few months ago, if anybody got a good idea (particularly one of our programmers), that person would just drop everything and work on that. Our original design document dealt exclusively with the story, so in a way, every feature we have is the result of feature creep. Every weapon we've added, with one or two exceptions, is the result of Jason, our modeller, getting a cool idea and rolling with it. Once he'd modelled, skinned, and animated a weapon, we'd divert resources to coding it up and fitting it into the game. That's very symptomatic of how our mod was developed for the first 4 years or so.

What sort of tools and programming have you been using for the mod, and how do you feel about these tools today? Would you have chosen otherwise today?

Sorry if I'm gonna go on a bit here. We've primarily been using the Deus Ex SDK. This notably includes a modified Unreal Editor and ConEdit, the original dialogue editor used for DX. Additionally, Jason uses Cinema4D to create all our models and 3DS Max to import them into the game. We've made a few tools of our own, too: Mostly simple applications like a line counting tool that helps determine the proper length for in-game text files. Finally, we use phpBugTracker (<http://phpbt.sourceforge.net/>) to keep track of all the issues we find during testing and development, and we have a custom web database designed to keep track of our voice acting. We've also been using the bugtracker for feature requests and such, even though it's not strictly best practice.

UnrealEd is simultaneously a great editor and a steaming pile of crap. The way it handles geometry and light calculation etc. is really clever, and its triggers and scripting integration makes a lot of sense when you know how it works. Unfortunately it also has some truly infuriating bugs, like a tendency to crash if you have a property window open without selecting any objects or an annoying habit of freezing if too many polygons are visible at once. For an editor from the last millennium, it's surprisingly logical and feature-rich, but ever since UnrealEd 2 was hacked to work with Deus Ex, we've been using that whenever possible (the

limiting factor being that UEd2 doesn't support Deus Ex' special model format, so any object or character placement must be done in UEd1). ConEdit is a great dialogue editor, but I have a feeling it wasn't designed for *quite* the level of interactivity we've aimed for in TNM. ConEdit is great for largely linear dialogue with a few choices and conditionals here and there, but if you want truly branching dialogue, you're better off with a more tree-oriented editor such as the dialogue editor from Neverwinter Nights. That said, we've done a few tweaks to ConEdit over the years, and it definitely serves its purpose now.

As for the programming, Unreal uses its own particular scripting language called "UScript", which is a lot like Java but with a syntax based on C. There are very few things you can't do with UScript – the main limiting factor is figuring out how the original game systems work. I have a lot of respect for the three ION Storm programmers who coded the whole game in 2 years, but the time pressure they were under has resulted in some frequently very hacky code, and it can be pretty tricky to modify without breaking. The main problem with modding Deus Ex (and, I suspect, most games) is that a lot of code is hidden from you – a lot of functions are "native", meaning they can't be directly edited. Thankfully one of our coders, Nick, figured out a way to "hook" into those functions and sort of override them with his own binaries (compiled with Visual Studio), so now we can do pretty much anything we'd ever want to do in TNM.

If I was starting up a new game project today, I'd clearly choose a more modern version of the Unreal Engine, such as the one used in UT 2004. I hear Hammer is pretty good too, but I'm so familiar with the Unreal way of doing things now, I wouldn't want to start all over if I could help it. The downside of using modern technology is that the detail level is so much higher – one character model in Unreal Tournament 3 is made of 2-6 *million* polygons, which is more than every level in TNM combined. Making a mod the size of TNM with the visual detail level of the Unreal 3 engine would take half a lifetime.

You have a lot of dialog in TNM, and according to your website everything will be voice acted. So I'm sure some people are wondering how you plan this process, as well as how you keep it all structured during the implementation into the mod?

Our voice acting process is so involved we've always been referring to it as a whole separate "department" of the development team. Lawrence and Gelo maintain contact with all our volunteer actors and manage the traffic of scripts and recordings going back and forth. Usually the process goes like this: First somebody sends us a random recording of their voice. We listen to it and send them some typical lines for two or three characters we think would fit their voice. Then they record these "audition" lines and send them back, so we can hear if we like their interpretation of the character. If not, we send them something else. If we like it, we collect the full script for the chosen character and send it to them. Once we get the recordings back, we upload them to our FTP server for future processing.

The processing itself consists of first cutting each line out into a separate file and naming every file to fit with ConEdit's audio file structure, and then uploading these files for our associated audio studio, BrandX (<http://site.brandxaudio.com/>), who are kindly processing all our audio for free. BrandX handles noise filtering, equalizing, effects, and volume control, and then they send the lines back to us so we can put them in the game. Usually this entire cycle takes several months. As I mentioned above, our good friend Kale coded up our own PHP and MySQL web database to keep track of the files. It's an extremely useful

utility which has an entry for each character specifying character name, conversation names, actor, status, and additional comments. Without this database we'd all have lost our minds trying to keep track of everything.

Deus Ex is a fairly old game today. Have you done any special effort to make TNM look more “up to date”, or will it look like DX when you release it?

We've pretty much pushed Deus Ex to the very limits of its graphical potential. Many of our maps are so large and detailed they can only be safely edited in wireframe mode or they'll crash the editor, the majority of our models have many times more polygons than the average Deus Ex weapon, and almost all the textures we've added to the game are exactly as large as the engine will allow – or even bigger.

Unfortunately when you edit Deus Ex you're working with some annoying restrictions you might not face with other games. Presumably because the latest Unreal Engine is still based on the very first one, Epic has never open-sourced their old work, unlike eg. id Software. However, there have already been several efforts to make Unreal and Deus Ex look better, and we've gone to great lengths to support some of them. Most notably, a new OpenGL renderer has been written, which allows us to use textures bigger than 256x256 pixels. Furthermore, we're working very closely with the well-publicized High Definition Texture Package (<http://hdtp.net/>) mod, who have remodelled almost all the decorations and weapons in the original game. If you play TNM without HDTP, our own weapons will probably stand out a bit because they have more polygons and bigger textures than the original Deus Ex weapons, but if you use HDTP (and why wouldn't you?) it'll even out.

Final question – When should we expect to see TNM released?

I prefer to do a Duke Nukem Forever and say “When it's done”. We've missed our first deadline by 5 and a half years, and we've made a habit out of missing at least one deadline per year since then, but I can tell you that right now, to the best of my beliefs, there's a good chance TNM will be done over the summer. It all hinges on our voice acting.

I would like to thank you for doing this talk with me for RazuBlog, but before we end do you have any final comments about TNM and its development?

Well I'd like to give out some advice to anybody who's considering doing a mod: Start out small, write up a thorough design document, and then *stick to it*. Feature creep is fun but unless you have 5 years to spare on your project, it will do you in. If you're organized, you can pull off any idea you might get, but if you're not organized you'll be throwing so much effort right out the window.

And then I'd just like to thank you for your interest in our project, I've been living and breathing TNM for the past 6 years, so I always enjoy any opportunity to share my thoughts about it. If you enjoyed Deus Ex, I'm confident you'll enjoy The Nameless Mod. And if you didn't enjoy Deus Ex, it's just because you don't understand it ;)