

The Nameless Mod Post-mortem

by Jonas Wæver

If we'd known what we were getting ourselves into, we'd have run away screaming. The Nameless Mod for Deus Ex stands out for a couple of reasons. First of all, it's a single-player total conversion, which isn't the easiest or most popular category of modification on the Internet. Secondly, it's enormous: The Nameless Mod features over 14 hours of voice-over, the average length of a playthrough is 15 hours, and that's just playing one of two significantly different, mutually exclusive plotlines. Third, and arguably most notably, The Nameless Mod has been finished and released to very favourable reviews. The project was in development for 7 years, 2 months, and 11 days by an international team of hobbyists who never met each other. It placed enormous demands on communication, leadership, and quality assurance, and to top it all off, it started as that most reviled of genres: Forum fan fiction.

Back-Story

The Nameless Mod for Deus Ex, released 9 years after Ion Storm's magnum opus, strives to reproduce the multilinear design and unique aesthetics that made Deus Ex such a widely beloved classic. While expanding on Deus Ex's core gameplay, the Nameless Mod also introduces players to an outrageous new world which is at once an homage to, and a satire of Deus Ex, its community, and the Internet at large. Playing as Trestkon, a respected veteran returning to his old Internet forum, the player is charged with investigating an apparent abduction which takes him deep into the history and politics of the community.

Game Data

Release date: March 15, 2009.

Genre: first-person action/role-playing game; cyberpunk spy thriller; pop culture satire.

Core team size: 1 producer, 1 lead designer, 1 dialogue manager, 2 programmers, 5 level designers, 1 3D artist, 4 2D artists, 1 sound designer, and 3 composers.

Additional contributors: 4 programmers, 6 level designers, 6 2D artists, 4 writers, 6 sound designers, 5 composers, 23 testers, and 74 voice actors.

Length of development: 7 years, 2 months, and 11 days from conception to release.

Notable technologies: Unreal Engine 1 and the Deus Ex SDK including UnrealEd and Ion Storm's ConEdit. One of our programmers added OGG music support.

The Concept

I would be lying if I said we had a game of TNM's final scope in mind when we started out in January 2002. We were originally planning a single "fan-mission" – one or two maps connected by a simple, quickly grasped plot, with the twist that the characters in the mission would be based on the users of GameSpy's PlanetDeusEx forums where the team was formed. It was a premise that betrayed our low expectations for the size of the audience we would be able to reach: we were just making this for our own community. However, the self-indulgent concept turned out to be a huge help in attracting qualified help. Everybody who joined got to be a character in the story, and who doesn't want to see themselves in a game? As the team grew, the scope of the project slowly grew as well, and the concept was expanded. We decided to take it even further, making the plot actually take place on the forums, in a virtual world inspired by the stories and the imaginary spaces we'd created for each other.

The most central part of the concept was never actually discussed nor given thought for the first large part of development – it was a given, an unspoken rule that we all agreed upon: we would remain loyal to the original Deus Ex gameplay. Unfortunately our project initially suffered from a lack of conscious design: As we didn't have a lead designer until a couple of years into the project (I was a dedicated writer to begin with), we had nobody responsible for analysing and recreating Deus Ex's gameplay, which is why our initial designs missed the mark in several important ways. Our first concept was designed to be one very large hub level with complete freedom to move to the connected maps and handle the missions in any order. A prototype of the hub map revealed several problems with the single-hub structure, greatest of which was Deus Ex's old Unreal Engine 1 and its hard limit on level size.

Through four revisions of the first design, we eventually settled on a design more closely matching Deus Ex's own structure: A largely linear plot with branches and variations along the way, taking you through several hub levels. Unlike Deus Ex, however, we chose to branch the main plot into 2 parallel storylines early on. We also chose to connect our three hub maps and distribute the missions across them all so the player could move freely throughout the game world with a single "point of no return" about half-way, giving us an opportunity to reset the game world and move time forward to show the effects of some of the player's actions. The result of these decisions was a game structured much like Deus Ex, but with far more freedom for the player to change the plot and far more optional content, but also a significantly shorter critical path.

The Production

Initial development was incredibly chaotic. We had no well defined team structure and no communication infrastructure to coordinate our work. Level designers began to build maps before the plot had been written, characters were created based on the online personas of the forum users and were then written into the story afterwards, and weapons and items were constructed on the basis of a cool idea with no thought as to how they would fit into the game. The design documents were retroactively written around game assets that were already being created, taking input from everybody on the development team and anybody else who cared to contribute on the forums.

Over the course of several years, we gradually got a handle on the project. By trial and error, we worked out a sane workflow and a reasonable team hierarchy, established reliable communication channels, and developed a solid understanding of our own project and the game it was based on. Whether by natural evolution or a lack of imagination, our final team structure was very flat, with our producer Lawrence Laxdal and myself functioning as the intersections of all communication. We became more demanding of our contributors and learned to quickly let go of anybody who didn't keep their promises. Moreover, we all became better at our respective crafts. Though this was a benefit for obvious reasons, in terms of scheduling, our perfectionism proved to be a problem. We estimate that 90% of everything we created in the first 2 years was later replaced, redone, or removed as our skills improved and we saw the flaws in our previous work.

When almost all our level assets were completed and we had a playable alpha, we took a good hard look at what we had done and what we still needed to do, and then we began to cut down on the feature creep. We've almost regarded feature creep as a form of currency with which we've paid our team – for every hour somebody has contributed towards finishing the game, we had to let them spend X amount of time working on their own pet features. As the project neared completion, we reduced X until the project

entered feature lockdown around the summer of 2008. By this time, only the most dedicated team members were left.

The final stage of production was dialogue audio and quality assurance. The process of recording the over 195,000 words of dialogue in TNM began several years before release and is notable for making use of a large amount of professional or semi-professional voice-over artists who contributed entirely for free. For this enormous project, we created an online database system which kept track of every character in the game, every conversation node attached to that character, each character's actor, the status of their lines, and which of our audio engineers was working on them or had worked on them. Early on, we recruited somebody specifically to oversee the voice-over process, and towards the last phase of the project, Lawrence and I ended up spending most of our time helping out as well. We always knew the large amount of volunteers would be a liability to the time frame of the project, but in the end we managed to wrap up the recording process about a month before we were finished with QA, much to our surprise. Three release candidates later, The Nameless Mod was finished.

What Went Right

1. Designing for the niche

It took us far longer than it should have, but we eventually figured out how Deus Ex worked, what made it great, and what we had to do to maintain its core gameplay. We made the decision early on to stay loyal to the original game; Deus Ex was very well designed and had a clear high-level vision, and understanding the design principles behind the original game allowed us to support and polish the existing gameplay and create a new setting and plot reflecting but not imitating Deus Ex's.

Judging from how TNM has been received, this was definitely the right choice. When developing a mod (or, one would imagine, an expansion pack or a piece of downloadable content or similar), it's a safe bet that the people who will be interested in your product are the hard core fans, and so it's not a bad idea to design your game for them. With this in mind, TNM had more freedom than Deus Ex and significantly more replayability. It also had a greater emphasis on role-playing, more long-term choice and consequence, and was a lot more difficult. None of these features help when it comes to accessibility, but to people who already knew Deus Ex's gameplay well, our design seems to have hit the sweet spot.

Furthermore, most of our design choices were aimed at tipping the balance of the gameplay elements further towards role-playing and adventure dynamics: Talking to NPCs who react to your play style, making long-lasting choices, exploring large and detailed environments, tackling problems with thorough use of your character's skills and abilities, etc. In other words, exactly what many role-playing fans seem to feel is missing from more commercially viable RPGs. It was obvious that people wanted more Deus Ex, and I believe we succeeded in delivering that, while adding our own stamp of uniqueness to the experience.

2. Turning a joke into a setting

One of our main concerns was how the scope and the aim of the project had gradually changed over the first couple of years. The Nameless Mod had started out as a small-time project aimed at a very limited audience, but after a year of development, we realised that we'd spent so much time and energy on the game already, we had to appeal to a greater audience - even though no money was involved, we felt that the more work went into this game, the more people needed to play it in order for our hard work to be justified.

Unfortunately, the game we were working on had a distinctly limited appeal: It was self-insertion fan fiction full of in-jokes, with a plot held together almost entirely by obscure gags and references to personal interpretations of events of no interest to anybody outside our small community. It was a mess, and it had to change. The primary way we did this was to bring people on board who had little to no attachment to our community and get their feedback. The fundamental causalities and motivations of the plot were rewritten, the setting was continually fleshed out and supported by explanations and player training, the fiction was tied together to make it more self-contained, the dialogue was repeatedly rewritten and edited to have fewer references and more character, and I did my absolute best to imbue the game with as much cultural relevance as possible, hoping people outside the development team would find it meaningful.

We started alpha testing relatively early, before the game was playable even half way, bringing in testers who had no knowledge of the Deus Ex community to get their general opinions and impressions about the game, and we improved the game in many ways with their help. The feedback we picked up immediately eased our worries: The game generally felt consistent, internally logical, and engaging. Nevertheless, we kept polishing the game as testing progressed and more people joined our QA team, realising that the unpromising concept of The Nameless Mod would be a hard sell. Judging from what chatter we've been picking up post-release, the general consensus seems to be that everybody hates the concept of the game until they actually play it, and that may be the greatest compliment we could get, in that we seem to have pulled off what most people considered impossible.

3. Managing complexity

Deus Ex was a complicated game with many different genre elements and many interacting and overlapping gameplay systems creating joyously unpredictable results in the hands of its players. Lead Designer Harvey Smith famously used the phenomenon of grenade climbing (utilising the fact that wall-mounted grenades have collision cylinders to scale a wall by jumping from grenade to grenade) to illustrate the dangers of emergent gameplay. We've always considered the benefits of emergent gameplay to be well worth this "undesirable emergence", and so one of our main goals was to add even more complexity to the game, both on the narrative level and the mechanical level.

In terms of gameplay however, we had an easy time compared to Ion Storm: Rather than having to invent an intricate web of game systems ourselves, we had the luxury of simply expanding on and tweaking a tried and tested set of mechanics. We chose to keep almost every object from Deus Ex (be it items, augmentations, or skills) and just add our own creations to this already sizeable collection. When we did change objects from the original game, it was to add further features to skills or augmentations that were widely considered to be less useful than their alternatives. We also had a lot of fun adding further layers of interaction to the game world, identifying such incidental interactivity (flushable toilets, readable emails, playable pool tables) as one of the great strengths of Deus Ex.

The real complexity came from our extremely ambitious plot design. I'll be the first to admit our ambitions in this area passed far beyond the boundaries of reason. One of the major points of criticism of Deus Ex was that you were apparently offered a choice of whether to leave an organisation, only for that choice to be made for you. I wouldn't call this a reasonable criticism, considering how much extra work such an option would've called for and how poorly it would've worked with the overall concept of the game, but we still decided to give the people what they wanted: two parallel, mutually exclusive storylines.

The key to basically creating two games in one turned out to be - perhaps unsurprisingly - reuse of assets. Early experience taught us that levels were the most expensive type of content, so we had to minimise the amount of unique maps per storyline. The cheapest resources turned out to be dialogue and gameplay scripting. Thus, we let most of our missions take place in maps that served other purposes as well, for example having one faction task you with breaking into the headquarters of the other faction and vice versa. By switching out all the plot-relevant dialogue depending on the storyline, but keeping the same maps, weapons, and characters (as either enemies or allies depending on the player's choices), we were able to create drastically different plotlines without too much unnecessary work.

That said, designing levels to be your home base in one case or a hostile mission area in another case, and writing and scripting characters to be either allies or enemies complicated the game logic more than was perhaps necessary, putting incredibly high demands on quality assurance - but I'll save that for the What Went Wrong section.

4. We released!

We actually finished the game after 7 years of work. Though many were itching to count us among the many over-ambitious total conversions that would never release anything before the developers eventually gave up and called it quits, we never stopped working on the mod. We had periods of low activity, but at any given time, *somebody* was always making progress towards finishing The Nameless Mod.

The fact that we released was largely down to three factors, chief of which was sheer determination to see this through. Though TNM started out as a pure hobby project, a couple of us realised along the way that game development was something we wanted, nay *needed* to do for a living, and we'd be damned if TNM wasn't going to be the greatest portfolio piece in the history of game design. There was also a pervasive sense that we'd come too far to give up on the game now - one does not simply sacrifice 4-6 years of free time on a project only to walk away from it when the end was in sight.

Even with this hell-bent determination to see things through to the end, we would never have finished anything if we hadn't started to reign in our own creativity. Feature creep was the engine that kept us running up to a certain point, but then it started to become a real problem - we weren't making nearly enough progress towards actually finishing the game, instead spending our time on whatever new features happened across our minds. Somewhere around 2006, we started making lists of what we had left to do and plans of how to finish it all. In other words, we became more responsible, more organised, and more professional. Eventually we entered feature lockdown, then complete content lockdown, and then we were done. Turns out the way to get things done is with good management.

5. Post-release support

After 7 years of development, most of us were really eager to put TNM behind us and start up a new, independent project - something we might actually be able to make money from eventually. Unfortunately that would have to wait. By the time we released, Deus Ex was a very old game, and in our infinite ambition, we'd been messing with the native code and adding a couple of our own features to it, such as an OGG music player - in short, it crashed *a lot*. Furthermore, it turned out that thousands of players are better at finding bugs than 20 volunteer testers (who would have thought?)

We started fixing bugs the day The Nameless Mod was released. When we built the release version, we were almost certain there were no significant bugs left, but hundreds of people signed up to our forums to

tell us otherwise. The first patch was released after one very long, very hard week of the worst crunch I've ever experienced, and fixed around 140 bugs. The second patch was released a month later with over 200 fixes. Though this sounds like a lot, it's actually fairly good considering the scope and complexity of the game, and the fact that we had very few really dedicated testers and no systematic testing procedures in place.

Most importantly though, we got them fixed very soon after release, with the help of our community, and we're not done with the game quite yet. We're also cooperating with a few other well known Deus Ex mods to help make sure their work is compatible with TNM. Finally, we've been providing thorough support to people with installation problems or other technical issues, going so far as to monitor TNM discussions on other forums to intervene with help when it's needed. All of this, combined with some very open and honest communication throughout, has earned us a lot of good-will with the community that we might otherwise have lost to a reputation for releasing a bug-ridden mess of a game.

What Went Wrong

1. We spent no time on pre-production

In stark contrast to Deus Ex's luxurious six months of pre-production, we literally had none. Actual game assets were being produced from day one; before we'd even decided what the game was going to be about or what sort of story we would write, artists were churning out character textures and level designers were building maps. A month or two of recklessly disorganised brainstorming segued into a confused, hurried documentation process where we struggled to keep up with our own ideas. Since the project was entirely anarchistic in the beginning, nobody had to wait for permission to start pumping out assets, and little to no coordination took place between different designers or artists.

In and of itself, "don't eliminate pre-production" may seem like a pretty obvious piece of advice, but what we were actually trying to do, in hind-sight, was iterative design: We were experimenting with the engine and the tools at our disposal, testing the capabilities of our team, and throwing our every idea at the game with no sense of restraint in order to see what worked and what didn't. The approach could've worked out a lot better than it did if we'd made a conscious effort to evaluate the decisions we made and deliberately pick out the elements we wanted. Our team size and structure was even quite suitable for an iterative design model; we had around ten highly dedicated people working on it back then, and there was no management or budget to worry about, so we were free to mess around and distil the useful ideas from the poor.

Unfortunately we made the major mistake of not eliminating anything at all. Everything people created and every idea anybody had went straight into the design documents. In that way, we managed to waste what was actually a pretty healthy pre-production process because we treated it as full production. A lot of the bad ideas and decisions that made it into the game back then, we managed to replace or fix later down the line, but some of it can still be found in the final product because it's simply too time-consuming to fix. File that one under "how *not* to do Agile development".

2. We didn't understand all the code

They say those who don't know history are doomed to repeat it, but sometimes you know going into a project that you're going to face some major problems, and there's very little you can do about it. The sometimes makeshift nature of parts of Deus Ex's code was one such problem we faced. Now, far be it from

me to criticise the Deus Ex programmers - they did an amazing amount of work and unlike us, they were working under deadlines. But by their own admission, the artificial intelligence in Deus Ex wasn't as well designed as it should have been and suffered even further for being constructed precariously on top of the Unreal Engine code.

To us, the problem wasn't just that we didn't want to mess too much with the AI code, and that we never had access to Unreal's source code to fix some of the more persistent bugs we encountered, but that we didn't fully understand how the code was meant to be used in the design. Our AI problems are the most illustrative example of that, but unfortunately not the only one. Deus Ex's AI, it turns out, is pretty stable when it just has to be friendly or when it just has to be hostile. When you start mixing friendly and hostile (and perhaps even neutral, if you hate yourself) behaviours in the same area, the illusion begins to fall apart. What we did was to mix friendly, hostile, and neutral characters in the same levels, and then switch their alliances around depending on the player's previous choices. It blew up in our faces, and it never stopped blowing up.

I suppose there were two ways we could have solved this problem: One was to sit down and rewrite all of Deus Ex's AI code as early as possible. This wasn't an option due to our extremely limited programming resources, so the remaining solution was to look very carefully at how Deus Ex's characters had their AI set up and how the different settings were used, and then stick to that; but once we finally realised that, implementing such a solution was tantamount to scrapping everything and starting over from scratch.

3. We had to compromise on quality

By the time TNM was released, our standards were pretty high. We screened our actors thoroughly for audio quality, we rebuilt entire levels dangerously late in development if there were too many problems with them, and we'd long since stopped implementing any feature if we didn't feel that we could take it all the way. But of course, beggars can't be choosers, and we sometimes had to compromise because we couldn't pay people. The two areas where I think this is most obvious in the final product are level design and voice-over. For the voice-over, it was a hard and bitter struggle to secure solid actors for the major characters, but I think we largely made it. In terms of minor characters, the picture looks a little different: Many of them were recorded by ourselves or our girlfriends or by random fans. Even our good actors occasionally had sub-standard recording quality, which becomes obvious far more often than it should.

In terms of level design, the problems were technical as well as aesthetic. Some of our level designers were quite good at producing pretty architecture, but didn't seem to grasp the design principles that made Deus Ex's missions so exciting. Others had brilliant ideas and a great understanding of design, but their levels looked boxy and uninteresting. Some levels that were otherwise nice and well designed had been constructed entirely off-grid, causing substantial technical issues later on, and it's painfully apparent that we never had an art director or a concept artist, making our levels vary erratically in style and tone. Far too late did we understand the concept of dividing level design tasks amongst a designer and an environmental artist, and even then, we never had the resources to do it in the first place.

Though this problem was emphasised because we were forced to make do with the generosity of volunteers, I imagine that too short a supply of talented team members who meet your standards of quality is a problem even commercial projects face, and I suspect that if I had a fool-proof solution for it, I could use it as the basis for a whole new article. In the future, we aim to conquer the problem by counting on a

small team of only the most skilled and dedicated rather than designing a project that demands the participation of 50 contributors of varying talent and enthusiasm. In short, we'll embrace the indie ethos and stop pretending we're an AAA studio.

4. We didn't test enough

We tried though. We gradually brought over twenty testers in to play the game starting several years before we were finished, and set up an online bug tracking system for them to report back to us with. We set up a forum specifically for the testers, and we sent out guide documents to everybody detailing how to install the test build, what sort of bugs they could expect, how to report them, etc. We evaluated our bug database and set up priority fix lists towards the end of the project to make sure all the really problematic bugs were fixed before release. When TNM was released, it had no major bugs that we knew about.

And yet the first journalist we sent the release candidate out to, before the download link was publicised, found a bug in the kill counting code which would close off an entire map prematurely due to a recent fix between the two last release candidates. We then spent an entire week fixing bugs in some sort of infernal post-crunch, followed by another month of slower, steady fixes. By the second patch, we've fixed around 240 problems, and we've more or less brought TNM into the state we thought it was in when we released.

There are three things we could've done to improve our quality assurance. First and most importantly, we should have been more careful to test our own work during development. Far too often, features and assets went into the game without any testing. In a game as heavily dependent on emergent gameplay as Deus Ex, it's critically important to apply very thorough testing to everything you implement: If we had tested each of our new additions in combination with as many different factors as we could think of, that would've significantly reduced the amount of combinations that slipped through the cracks. Secondly, we should have set up some auto-testing: One of our programmers coded up auto-testing scripts a couple of months before we were done, which was far too late - we only managed to apply it twice before release. Since our plot had so many branches and variations along the way, being able to fast-forward through certain common branches would've been a great help earlier in the project.

Finally, we should have managed our testers more strictly. Unfortunately I'm not sure most of our testers would've responded very well to tighter management, but I suspect five dedicated and well managed testers would've been far more useful than a score of testers left mostly to their own devices. We tried to get our testers to commit to particular kinds of playthroughs, hoping we'd get all the branches covered, but it never worked because most people who volunteered to test weren't interested in committing to anything beyond playing the game and writing down the major bugs they found. In summary, our primary mistake was that we skipped the organised alpha testing and went straight to beta testing.

5. Our project is a PR nightmare

The final and perhaps the hardest problem we faced was making people understand that our game wasn't shit. We'd spent a lot of time making sure that people wouldn't be alienated by our game and that we had plenty of depth and internal consistency to keep everybody interested, but we'd completely underestimated what a huge turn-off the basic idea of the mod was to so many people. Every site where we published our trailer, and every forum thread where people began to discuss the mod, one sentiment would immediately surface like a knee-jerk reflex: *What an idiotic concept. Why would anybody spend seven years working on this fan-boy circle-jerk of a game?*

We were pretty crest-fallen: We'd gone to such lengths to make sure TNM was a game, not just a joke, and many people wouldn't even give it a chance because they immediately assumed the worst. But to make matters worse, we came to realise that we'd frontloaded all the Internet references, the fan culture and the memes and the in-jokes right in the first mission of the game. Part of this was unavoidable: The first mission served by necessity to introduce the player to our setting, so all the opaque references and Internet semiotics were presented to you immediately. Once out of the first introductory hub area, the setting would quickly slip into the back seat to leave room for the plot itself, but too many people seemed to never reach it, having lost all interest long before then. Since The Nameless Mod is free to download, we have no demo, but in terms of convincing people to invest their time in playing through TNM, that introduction area is all we have, and it seems to be doing a rather poor job.

Perhaps our greatest mistake was to tell people that The Nameless Mod was inspired by a real community that existed on the Internet at one point in time. I suspect people in general would be a lot more susceptible to our quirky cyberspace setting if they thought we'd just invented it as a Snow Crash-esque sci-fi take on the Internet, because then they wouldn't associate us with the reviled genre of "forum fan fiction" to begin with, and once playing the game, they wouldn't be expecting in-jokes everywhere. Much of the feedback we've received has implied that people constantly see in-jokes and obscure references when by far most of the game's fiction was either invented specifically to suit the plot or the setting or twisted so far out of its original shape that it no longer bears any resemblance to the events or the people it was inspired by.

A well-known games journalist graciously defending our concept wrote: "no wonder everyone makes games with space marines being gruff". I'd like to think people are generally open to new concepts, but I'll admit that our premise and our setting do us no favours. Not, however, because TNM is too weird - there have been many successful games far stranger than The Nameless Mod and our world is quite recognisable when it comes down to it. But at the same time it reminds people of a genre of fiction which is almost never executed well. If we'd managed to set ourselves further apart from that genre with all the PR material we sent out, and if we'd done more to change people's expectations before firing up the game, I think we'd had a much easier sell.

In Closing

In some ways, The Nameless Mod is a triumph for Agile development and iterative design processes. Not only did we learn how to make the game we were trying to make, we learned almost everything we now know about game development and project management. We learned these things by trial and error, and our errors pushed us to seek out the information we needed and apply it to our own work.

It's unfortunate that we wasted so much time messing up and making bad design decisions. Part of this was because we didn't manage feature creep as well as we could have, and part of it was because we got carried away with perfectionism, constantly recreating old assets to match our new standards. If we'd known what we were doing right from the start, TNM could probably have been released in 2006.

It's been an incredible project though, and it's changed some of our lives substantially. I think we ultimately came out on top of most of our mistakes, and we're very excited to move on to the next challenge. With all the feedback we've received on The Nameless Mod, hopefully we won't repeat any of the mistakes we've made. And now we have this post-mortem to help us remember them.