

passing
through
larceny

a Neverwinter Nights 2 module by Richard Ericksen

about this module



Passing Through Lorren is a module I've created to demonstrate, basically, that I can create a module. I'm looking to break into the gaming industry, specifically into making MMOs. I've written up documentation on ideas I've had on boss encounters or specific facets of gameplay in general, but have been told that the best way to focus my energies is by grabbing a toolset and showing that I can put the tools to work to back up my writing.

I've worked professionally with computers for years, and feel that a strength of mine is the ability to just grab a new program and start flipping levers and twisting knobs; digging through help files to figure out what does what. Learning a new program is a huge turn off to some people, especially if they resent that the new version made most of what they just struggled to learn in the previous version obsolete. I'm the guy that's eagerly downloading new betas to see what cool new stuff they've packed in. I love trying to find multiple ways to do the same task using different tools at hand, and I approach difficult situations with the knowledge that there's usually a better way to go about it.

Some quests included in this module are of the vanilla MMO 'standard fare'. Kill ten rats, go fetch five spores. These portions of the module are me sitting in the driver's seat of a car during a driving test. It's where I demonstrate that I understand about coming to a complete stop at a red light, and looking both ways before proceeding. I get the basics. The creative side (I hope) is where I've branched the plot according to how you go about the *handling* of the spore fetching. The moral decisions made, and whether you 'stick to the rules' or not are what determine which of the ten endings you arrive at.

The basic task at hand is to catch a boat home from the harbor. There are many different ways to go about solving that problem. I've tried to leave as many *hows* in the hands of the player to run with as I could. This booklet is essentially a huge spoiler, so I hope you take a short time to fire it up and run through it once or twice before returning to the dissection of the parts I'll be laying out here. It's not a horribly long play; I actually kept it rather short on purpose. The main point is the replayability offered by the multiple branching plot paths.

For those looking for a walkthrough, the back cover to this booklet breaks down how to pull off each ending.

objective



In a single sentence: You're an outsider passing through town on your way east to catch a boat home, and you need to come up with 100g to pay for the ride.

I've intentionally left a blank slate for the player to work with concerning the main protagonist's background. I did, however, include a specific character for them to play with, because certain plot points reference them being non-human (the included character is Drow). Too much provided background would influence the player, and I wanted the player to make their own decisions on how to approach certain tasks. Travelling nuns don't stab people in the back, and escaped lunatics aren't likely to lend a hand at bake sales. By letting the player fill in their own story they can feel like there isn't a 'correct' path to take at any given moment.

The intro and ending movies are pre-rendered clips that show the protagonist as pictured above, as well, so if you rolled Frodo it would be awkward. Given that your character is a female Drow, on her way to the harbor, how do you proceed to come up with the cost of a ticket?

the area and history



The twin towns of Lorren and New Brill share a history. Lorren is the last stop on the way east towards the harbor town of Pinole, and first stop going west for Pinole's new arrivals. At first, it was only Lorren serving the trickle of people passing back and forth across its gates, until a sailor named Johannes decided to settle there after retiring. He discovered there were mineral deposits found across the river to Lorren's northern border. Cap'n John, as he was called, began to clear the area and set the foundation for what would eventually become New Brill.

As the mine matured, and more people settled in the area, New Brill became a town in its own right. Its location, like Lorren's, allowed easy access to Pinole's harbor. Fresh ore could be easily shipped off in boats across the sea to the east, and new arrivals headed west in need of supplies could find freshly crafted goods in the twin cities.

All was well, and the two towns grew and prospered for three generations. Tragedy struck, though, in the form of a deadly virus that devastated the people of New Brill.

Miners and their immediate families were the

first to become infected, and it was the late Cap'n John's own great-granddaughter that was the first casualty. Her father was foreman of the mine, and brought the sickness home with him in those first, frantic days. Ysharros, the woman, was married to the local doctor, a Drow by the name of Tesh. He worked frantically to discover the root of the problem, but soon became overwhelmed as the entire town fell ill.

The Miner's Virus, as it soon became called, spared no one in town. The doctor himself, being Drow, was found to be immune. Initial infection to time of death took mere days in each case, and Tesh watched helplessly as everyone he knew in town died painfully to the sickness. Across the river to the south, in Lorren, hasty barricades were built, and a quarantine was put into effect whereby no one from New Brill was allowed across the river for fear of spreading the infection. Tesh and his half Drow son, Mordiceius, would come into Lorren from time to time for fresh supplies, or to confer with the local mages that ran a store in town called Word of Shadow.

One day, on such a trip across the bridge,

Tesh was detained against his will and held in Lorren. The mayor of Lorren, Tobold, had decided that New Brill was a lost cause, and that Tesh was needed in Lorren in case the infection should crop up there. Tesh wasn't happy about being told to abandon his home in New Brill, but the fact remained that the entire town had succumbed to infection, and Tesh had made no progress as of yet.

Tesh and his son were set up in Lorren and eventually the last residents of New Brill passed away. The order came down from Tobold to have Spinks, the local law, set the town ablaze in hopes of eliminating the virus. The town burned, but the virus remained, visibly clouding the air in a thick haze.

A week passed, and Tesh sequestered himself in his new home overlooking the center of Lorren. People let him be; the burden of mourning an entire town that you provided help for is no easy burden to carry. Unbeknownst to the people of Lorren, though, Tesh's son was sick, and he was frantically renewing his search for a cure.

Tesh's only contact with the people of Lorren were the two mages in town, Melf and Crimson Starfire, that ran the Word of Shadow next door to the Pink Pigtail Inn. They brought him food and supplies, and discussed possible directions of study for the virus cure, should new cases arise in Lorren. One day, they found Tesh's door bolted from the inside, and they, too, gave him space.

Tesh's son, Mordiceius, eventually fell to the virus, and Tesh, in his desperation, resorted to drastic measures to preserve his son's body. Upon finally making the breakthrough he needed, his son had been dead for two days. Tesh had drawn an arcane pattern on the floor to contain his son's soul, and administered the cure to his son before undertaking a ritual to bring him back from the dead.

The ritual was taxing, and as the final incantations were read, Tesh collapsed on the floor of his hut, blacked out from exhaustion. Everything had come to a head, and his body couldn't bear the strain.

Mordiceius rose from the ground, cured now, but undead, and shabled from the house, down into the town below. Spinks came across Mordiceius in the hours just before dark, and found him hunched over something by the Pink Pigtail's stable area. It was Mordiceius, chewing on freshly slain horse's leg. He rose and ran at Spinks, and she cut him down easily. A crowd began to gather from within the bar, and no one saw Tesh running down the hillside until he was there on the ground beside his son. He scooped his son up in his arms, and cursed the townspeople before running north across the river with his son's corpse.

The townspeople couldn't follow, for fear of infection, and many figured Tesh wouldn't be very interested in anything they would have had to say regardless.

Tobold personally cleared out Tesh's temporary home, discovered the pattern drawn on the ground, and realized that Tesh took his notes with him when he fled.

While the people of Lorren felt as though the virus had all but run its course, they were alarmed to realize that Melf and Crimson now carried the infection. Mordiceius wasn't fully immune, being only half Drow. The virus they carry seems to be a slower working mutation of the original, but without help soon they'll suffer the same fate as those in New Brill.

Meanwhile, along the road headed east, a young Drow woman named Ixobelle is making her way to Pinole Harbor, and looking for a way to come up with a way to pay an eastern fare as she passes through Lorren...

cast

The supporting cast to *Passing Through Lorren* is comprised of familiar characters if you stay abreast of WoW or MMO blogs at all. My rise to supreme internet superstardom* has been well documented at ixobelle.com, and before that in the annals of the great parody site NotAddicted.com (*'The softcore site for hardcore casual gamers'*). NA is a shambling mess these days after some half cocked 'site overhaul' had it offline for 6 months or so. The only comments these days are from spam bots hawking replica Louis Vuitton or offering tips on how to get your manhood back into action. My fellow bloggers have been with me through the thick and thin, and cheering me on for this wonderous journey, though, and this was a little wink and nod fanservice thrown their way...



Hatch esc-hatch.blogspot.com

Hatch is the new doctor freshly arrived in town. He's a bit confused about the whole situation, because no one in town seems very eager to talk about the whole mess with Tesh. He just wants to find a cure and help the sick.

Melf and Crimson Starfire wordofshadow.blogspot.com

Two mages who tried to help Tesh come up with the cure, they became infected through close contact with Mordiceius, Tesh's half-Drow son that everyone thought immune. Together they run the Word of Shadow, a magic shop located next door to the Pink Pigtail Inn.



Larisa and Elnia pinkpigtailinn.com

The pint sized but feisty owner of the Pink Pigtail Inn, Larisa, and her trusty bartender, Elnia. Elnia is a quiet observer, and can be counted on to give an impartial history on the town, or keep her ears open for any odd job leads that she might overhear at the bar.

Jong and Megan dpspaladin.blogspot.com

Jong is the greatest paladin in the known universe to ever swing a two handed *anything*, and his maiden in waiting, Megan, writes ballads to his glory while looking wistfully on. I was torn between leaving my original conversation dialogue I wrote for Jong in or not... it was all pure l33t speak/Barrens chat with lots of ones and exclamation points, but in the end I just made him overly dramatic. It still works, I think.



**supreme internet superstardom subject to debate*



Gevlon *greedygoblin.blogspot.com*

The local merchant, and resident obnoxious ‘greedy goblin’. He’s looking for someone to eliminate the one source of competition he has in Lorren, the blacksmith Syp.

Syp *biobreak.wordpress.com*

The town blacksmith, he also runs a small shop in town. He has an unusual tendency to craft nothing but shovels for some odd reason, and has reason to believe that Gevlon has been stealing his precious shovels under cover of night. He’s feeling the pinch of being cut off from his source of raw ore.



Tobold *tobolds.blogspot.com*

The mayor of Lorren, he’s the first to tell you that everything he does is for the greater good of the people. That may actually be true, but his hamfisted approach to most situations doesn’t tend to go over well with those usually affected by his ‘goodwill’.

Spinks *spinksville.wordpress.com*

The local law, and a retired knight, she takes orders from Tobold. Her life has been one of endlessly obeying her superiors without question, but recently she’s had enough with Tobold’s grand solutions. She offers help when we need it, but only grudgingly.



Tesh *tishtoshtesh.wordpress.com*

The previous town doctor, and a Drow, like yourself. He was found to be immune to the Miner’s Virus, but it didn’t help all of those around him from dying. Tesh’s beloved human wife, Ysharros, was the first to go, followed by every citizen of New Brill shortly thereafter. He spiraled out of control when he realized he was losing his son as well.

Mordiceius *mordiceius.com*

Tesh’s half Drow son, whom everyone thought immune. His infection took longer to take hold, and was perhaps a mutation of the original strand. After dying, he was resurrected (and cured), before being struck back down again as an undead by Spinks.

Not appearing ingame, but featured in its lore:

Ysharros *stylishcorpse.wordpress.com* - Tesh’s lovely (and dead) human wife

Cap’n John *capnjohnsblog.blogspot.com* - The founder of New Brill

Dialogue

I struggled for a good five minutes to not have the first sentence of this section be ‘the dialogue speaks for itself!’, and even now I have to live with knowing that I’ve failed. I’m a horrible person. I’ll go stand in the corner for a bit.

In all seriousness, writing the dialogue was one of the more cathartic parts of the module’s creation, and was something I would bang away at when I just wanted to make solid headway. I knew much of what I needed to say by knowing the storyline itself, and when I would sit down in the dialogue editor and flesh out certain parts of moving the plot forward, it gave a good sense of progress.

The dialogue for a module like this is a living thing, in that you’re going back and adjusting the tone of certain parts, or having to make sure what Tobold said lines up with what Hatch just mentioned. Sometimes I’d go into the editor, and what would come out would surprise me, and I’d need to adjust other people’s lines to jive up. Spinks, for example, is really pissed at Tobold when she gives us the work order for Quiet Death.



I love having the characters developed in my mind to the point where they speak their own lines. Sometimes what they say shocks me, and I need to go back and adjust the more flexible characters around them.

I love writing; this whole booklet is proof of that, I hope. It’s frustrating to know that out of the 10,000+ words of dialogue I wrote, only a tiny fraction will be visible to a player, depending on which route they take to the end. The conversation above only appears in one of the ten endings. Likewise, in an MMO, text is a lost cause. So many people will click through it as fast as they can, and then open the journal for a summary of what they just skipped.

No sense crying over spilled milk, though.

For Lorren, I intentionally didn’t go Super Deep RP Mode for the text. Ixobelle talks like a modern person does. Somewhat modern speech like ‘*whatever*’ or ‘*gotta run, bye*’ come up from time to time. Other characters have more rigid speech patterns, and stick to them, or are even put off by our (Ixobelle’s) directness. Jong was originally 110% BaRR3NZ Ch4tZ0r\$, but it was just too much, so I revamped him to be just ridiculously over the top silly.

the quest journal

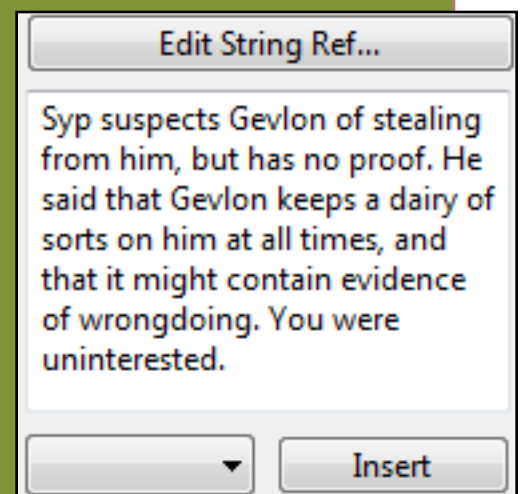


Many of the quests in Passing Through Lorren are pretty straightforward. I've made extensive use of the journal to make sure the player always knows what's expected of them, and when they have the option to return to a quest giver for something they've turned down through dialogue.

I've tried to keep options open for the player to do what *they* feel is important. Nothing is more eye-glazing than being herded down a path you aren't interested in walking. I tried to break up any long conversations with plenty of 'you know what, I really just don't care' options for the impatient, leaving subsequent summaries in the journal. The very first thing that happens in the module is that Hatch runs up and begins talking (starting the player down the main, good, path), but the player is always given the option to ignore what's being told and just poke around. To make sure they don't feel left out in the cold, the quest 'waits' on a side ID number, and if the player decides to go back later they can pick up from that point.

I've tried to include this for any quest where it's appropriate.

Other quests involve looting specific objects or receiving items like bar tabs from certain NPCs. At first, the implementation of these was clunky, I had generic "bar tab" items that would be tracked, but the quest journal wasn't giving very useful feedback to the player. "Two of three tabs collected" meant that the third could be anywhere if the player came back from a save game, and wanted to pick back up in the middle of a quest. I solved this by assigning each tab a static number weight; the quest log was suddenly able to effectively report which tabs had been collected, and which were still outstanding.



It was a borrowed form of binary that was the key to what I wanted to do:

- 001 (1) = tab one (Tobold's)
- 010 (2) = tab two (Syp's)
- 100 (4) = tab three (Megan's)

Once the foundation was laid, it was a simple matter of just creating enough quest states to satisfy any combination. With the (accepted) quest ID being 10...

I'll Gladly Pay You on Tuesday...	
..... Larisa has asked you to collect outstanding tabs. You refused.	1
..... Larisa has asked you to collect outstanding tabs from some of the re...	10
..... You've collected Tobold's tab, but still need Syp's and Megan's.	11
..... You've collected Syp's tab, but still need Tobold's and Megan's.	12
..... You've collected tabs from Tobold and Syp, but still need Megan's.	13
..... You've collected Megan's tab, but still need Syp's and Tobold's.	14
..... You've collected tabs from Tobold and Megan, but still need Syp's.	15
..... You've collected tabs from Megan and Syp, but still need Tobold's.	16
..... You collected all the tabs! Return the money collected to Larisa at th...	17
..... You returned the collected tabs to Larisa at the Pink Pigtail. She gav...	99
He's Crafty	
..... Syp needs ore from across the river to craft some shovels. You polite...	1
..... Syp needs ore from across the river to craft some shovels. Go north t...	2
..... You got a chunk of ore, but still need two more.	3
..... You got two chunks of ore, but still need one more.	4
..... You collected the three chunks. Way to go, you chunk collector, you!	5
..... Syp was pleased at the prospect of building another ... shovel. He pai...	99

...we just start filling in the blanks. $10 + 1 =$ Just Tobold's collected. $10 + 2 + 1 =$ Tobold's and Syp's have been collected. Any combination can be expressed, and a corresponding entry created. With a larger six tab collection quest the same principle works, but it's just a lot more entries to create. You can see the next quest shown above is a much simpler "go grab three chunks of ore" quest that doesn't require any nifty tricks, but the same level of feedback in the journal lets the player know exactly how far they are in the overall scheme of things.

Figuring things like this out, and having an interesting puzzle to solve, was easily my favorite aspect of creating the module. The power you have behind the scenes --knowing that *the player* will only see a single line update in their questlog, but knowing that *you* went in and figured out that every quest state needed to be accounted for and manually entered in advance-- was a close second. If the player never realizes, or never stops to think about these kind of things, then you've done a good job.

Having that eureka moment when you understand how you can make it work, then putting it in there, running it, and seeing that it *does* work... clean execution literally *feels good*. One of the most jarring things that can happen in a game is when you bonk into something unintended and see behind the curtain as a player. Even worse is when it's obvious to you how it could have been done right, but the people behind the curtain didn't bother to get that ironed out.

All in all, though, the quest journal is pretty straightforward, and mostly just chugs along on its train tracks. Where it gets interesting, though, is scripting!

Scripting!



```
51  if (GetFactionEqual(oGuy,GetFirstFC())) //special cases only for PC-related stuff
52  {
53  }
54  // Teah's Notebook (q_fetchnotes)
55
56  if (GetTag(oItem)=="teashnotebook")
57  {
58  if (!GetLocalInt(GetModule(),"Teah's Notebook Obtained"))
59  {
60  if ((nQuestLevel=GetLocalInt(oGuy, "IN_JOURNAL_ENTRY" + "q_fetchnotes")) >= 2)
61  AdvanceQuest("q_fetchnotes", 11, oGuy);
62  SetLocalInt(GetModule(),"Teah's Notebook Obtained",1);
63  }
64  }
65  // Bar Tabr (q_ppitabr)
66
67  if (GetTag(oItem)=="tobolds_payment")
68  {
69  if (!GetLocalInt(GetModule(),"Tobold's Payment Obtained"))
70  {
71  if ((nQuestLevel=GetLocalInt(oGuy, "IN_JOURNAL_ENTRY" + "q_ppitabr")) >= 10)
72  AdvanceQuest("q_ppitabr", nQuestLevel-1, oGuy);
73  }
74  }
```

Scripting gets an exclamation point up there on the header. The quest journal is interesting, writing thousands of words of dialogue is a good challenge, but scripting, woo!

This is where the power is, and where you can make anything happen. I *get* that. My first foray into scripting in the NWN toolset was using premade *ga_* scripts that were attached to dialog nodes. You pick the script in a dropdown list, fill in some variables, and pew pew run the module. Did it work? No? Why not? Who knows!

It was horrible. Thankfully, it didn't last long.

I began digging around and learning nwscript. The more I poked around and looked at other scripts, the more it all made sense. By the end I was writing full fledged scripts on my own that will *probably* look lame to someone who's been doing this for 10 hours a day since the dawn of time, but I'm really very proud of what I did. My module *works*, and that's what's important.

Scripting is one of those aspects of implementation that just needs some time and experience to get the hang of. One of the biggest draws for me, as someone bent on efficiency, is that you don't need to reinvent the wheel every. time. you. write. a. new. script. Much of your previous code can be reused with minor modifications, and you can look at other people's scripts for inspiration once you *get* it. That, and that there's always more than one way to solve a certain problem. I love that.

The fact is that a huge portion of the modding community is terrified to sit down in the script editor and learn something new, or they rely too heavily on the premade scripts or these script generator things that have you push little buttons *beep boop beep*, and it spits out some un-commented mess in the end for you. I've worked with computers for years, in many different aspects, but I never sat down and just *wrote code* aside from login scripts, or some very basic HTML or PHP. Most of that was just web page layout, not fetching variables and the like. Now that I've done it (and *can do* it), I'm not sure what the big deal is.

Diving headfirst into the scripting proved to be one of my favorite parts of making the module, and I can only see it getting better as I grasp more advanced concepts and have others around me to bounce ideas off of. This section is basically a run down on a few unique problems I ran into, and how I handled them.

Incrementing the Quest Journal

As mentioned before, I had a few quests that needed me grabbing items or delivering things to advance quests. One of my longest scripts is the `o_mod_acquire` script that fires whenever an object is looted or otherwise acquired. I didn't write the whole thing from scratch, I took the existing `o_mod_acquire` that was already in the game, and added what I needed to be handled for Lorren specifically into it. Things like the bar tabs I mentioned in the quest journal section. The three bar tabs have unique number 'weights' to them (Tobold's tab = 1, Syp's = 2, Megan's = 4). This allows me to stagger journal entries and write a unique ID for each step in the quest ($5 = 4 + 1 =$ 'you have collected Megan's and Tobold's, but still need Syp's').

```
object oPC = GetModuleItemAcquiredBy();
object oItem = GetModuleItemAcquired();
int nQuestLevel;

if (GetTag(oItem)=="syps_payment")
    if (!GetLocalInt(GetModule(),"Syp's Payment Obtained"))
    {
        if ((nQuestLevel=GetLocalInt(oGuy, "NW_JOURNAL_ENTRY" + "q_ppitabs")) >= 10)
            AdvanceQuest("q_ppitabs", nQuestLevel+2, oPC);
        SetLocalInt(GetModule(),"Syp's Payment Obtained",1);
    }
}
```

... it's just a simple way to detect that his payment was received, and (assuming we're at least on quest ID 10 or above, and nothing wonky is happening) it grabs the quest ID and adds 2 to the number. This means I don't need to manually set the quest ID, I just increment it by the given value, and it all lines up because I manually wrote all the quest journal entries. I have similar entries for Tobold's and Megan's tabs.

This kind of thing was very common. A variation was when the plot branched depending on how a certain situation was handled. It took me a while to realize that having duplicate items was a perfect solution to this problem. I created two identical items, but tagged them as `<item>_peaceful` and `<item>_violent`, or `<item>_looted` and `<item>_pickpocketed`, and had the `o_mod_acquire` script take over when either version showed up in our inventory.

Here we have Tesh's missing journal page:

```
//Talk to Tesh, get page (q_thereandback) PEACEFUL

    if ((nQuestLevel=GetLocalInt(oPC, "NW_JOURNAL_ENTRY" + "q_thereandback")) > 1)
        if (GetTag(oItem)=="missingpage_peaceful")
            AdvanceQuest("q_thereandback",80,oPC);

//Kill Tesh, take page (q_thereandback) VIOLENT

    if ((nQuestLevel=GetLocalInt(oPC, "NW_JOURNAL_ENTRY" + "q_thereandback")) > 1)
        if (GetTag(oItem)=="missingpage_violent")
            AdvanceQuest("q_thereandback",90,oPC);
```

The quest *There and Back* (`q_thereandback`) has two separate endings depending on if you bashed Tesh's face in and took the page, or tried to reason with him peacefully. This could have been handled by whether or not he was killed, but that proved troublesome. What if you talk to him, and get the page peacefully, and *then* bash his face in? Another script destroyed the lootable version he carries when the peaceful version was handed over.

Things like this, scripts leading to other scripts, is where I began to realize just how far I could run with all of this. Where I realized how deep the rabbit hole can actually go.

Usable items

Another common scripting use is items that have specific purposes when clicked. In Lorren, these range from what I call simple *complain items*, where Ixobelle complains about something when clicked (*‘that looks too heavy to carry, no thanks’*), or NPCs with simple barkstrings, to more intricate items like Tobold’s guest bed located on the second floor of his house:

```
// tobold's guest bed use script

// basically, I want this to only be 'really usable' if we've been
// told that "we should go take a nap" by Tobold to advance to nighttime,
// where q_blunt and q_mercykillings have their grand finales.

#include "ginc_transition"

void main()
{

object oPC = GetPrimaryPlayer(); // define the player character
string sMessage = "I'm not feeling very sleepy right now."; // our complaint that we aren't sleepy

int nBlunt = GetJournalEntry("q_blunt", oPC); // defined this integer to grab the current state of q_blunt
int nMercy = GetJournalEntry("q_mercykillings", oPC); // defined this integer to grab the current state of q_mercykillings

//now the IF/ELSE statements that determine whether the bed is usable or not

if ((nBlunt == 2) || (nMercy == 2)) // if on the quest, jump to the 'night time' zone
{
    FadeToBlack (oPC);
    AssignCommand (oPC, JumpToLocation(GetLocation(GetObjectByTag("wp_night", 0))));
}
else
{
    FloatingTextStringOnCreature (sMessage, oPC); //otherwise, complain you aren't sleepy
}

}
```

The bed was an interesting one to figure out, but in a roundabout sort of way. A few of the module endings have you simply killing the infected of Lorren (oh jeez, spoiler, don’t read that last sentence). But (!) I wanted there to be a transition to night time for when it happened. I figured you wouldn’t just waltz up to the sickhut during broad daylight and slit throats, then happily march out of town. The whole thing is also Tobold’s idea, and he wouldn’t want anyone to know that he’s a part of it, so he asks you to go rest upstairs in his guestbed until nightfall, then return to him to carry out the task.

I rigged the time in the game so that it actually moves extremely slowly. The module shouldn’t take very long to finish, and I didn’t want to hassle around with shops closing and NPC night time tasks (irrelevant to the storyline, stay focused!), so I basically made it take two real hours to pass one game hour, which would mean you’d be sitting around for an awful long time waiting for nightfall. I also wanted there to be torches lit, crickets chirping, no NPCs milling about... you get the picture. What I basically wanted was a completely separate area, so I went ahead and made that. I duplicated my main town area, and started erasing things (people), and adding others (crickets, torchlights). This proved to be the best solution, as I had a night version of Tobold (with his own night conversation) and a night version of his house as well.

I dropped a waypoint next to his night version of the guestbed (where you ‘wake up’ to), and made the daytime bed a door (in essence) that transitioned the player to this night world. The script checks to make sure we are on either *q_blunt* or *q_mercy* step 2 (they both take place at night), then lets us use the bed. Otherwise the object is a standard complain item, and we gripe about not being sleepy (this comes up if we kill Tobold, take his 2nd floor key, and make our way upstairs without being on those quests). Oops, spoilers again...

NPC actions and behaviors

One of my favorite script sets is what I call Fight Club. This was perhaps the trickiest issue I solved, and I'm really pleased with how it worked out in the end. In NWN2, there are factions that NPCs belong to, and you adjust standing with factions depending on your actions. In Loren, there are two rival merchants, Gevlon and Syp. Gevlon asks you outright to whack Syp, and while Syp doesn't come right out and say it, he wouldn't miss Gevlon were he gone. These two have stalls set up in town, and I needed a way to make them both be separate factions, while still being aligned. I wanted guards to attack me if they *saw* me taking swings on *either* of these guys, but not maintain a grudge (if they *didn't* see me attacking anyone, but I bumped into them later), so the standard Merchant and Defender factions were out. I also needed Gevlon and Syp to revert to neutral with me after I killed one or the other, for quest redemption.

This all came down to creating Fight Club, a new faction that I made Gevlon, Syp, and the three guards in town members of. Basically, when I attack one member, they yell to anyone within earshot. If anyone hears, they'll come running and join in. Once I kill someone, there's a short pause, and then a check is run to see if I'm still in combat. This is in case I'm still fighting a Fight Clubber while one dies... we can't reset the standing to neutral in the middle of trading blows, so a short pause needs to be given. I (eventually, through debugging) realized this pause needed to be at least 6 seconds, since the game engine's heartbeat script that fires every 6 seconds is what determines whether or not you're still in combat. So I made the check happen 8 seconds after a death, just to be safe.

So let's say I engage Syp, and he calls a guard that's walking close by. I'm fighting both of them, and another guard waltzes through on his walk path. He sees us fighting, and joins the fray. I kill the first guard, and after 8 seconds a check is made to see if I'm still in combat. I am. The second guard and Syp are still hostile, and we continue to fight. I kill Syp, and the check occurs again while I'm still in combat with the last guard. Finally, he dies, too. I drop combat after the heartbeat fires, and the check is run. I'm not in combat, so my Fight Club faction aggro is reset to neutral, and I can happily walk past the last guard without fear of him attacking me. More importantly, I can report to Gevlon that Syp is dead (and collect on the contract).

Figuring out how to execute this correctly (the delay, and figuring out it was tied to the heartbeat) was the hardest part. In actuality, the scripts are ridiculously simple. I needed to use some debugging text to see when it was firing, but once I got the timing, I commented it out:

```
void main ()
{
    object oPC = GetPrimaryPlayer(); // us, ixobelle
    string sIC = "I'm still fighting."; // in combat
    string sOOC = "I'm not fighting."; // out of combat

    if (GetIsInCombat(oPC)==FALSE)
    {
        AdjustReputation(oPC, OBJECT_SELF, 50);
        FloatingTextStringOnCreature (sOOC, oPC); //debugging "say you aren't in combat" text
    }
    else
    {
        FloatingTextStringOnCreature (sIC, oPC); //debugging "say you are in combat" text
    }
}
```

The check is just one line that pauses for 8 seconds and then calls the above script. Simple!

```
void main()
{
    DelayCommand (8.0, ExecuteScript("dont_talk_about_fightclub", OBJECT_SELF));
}
```

miscellany

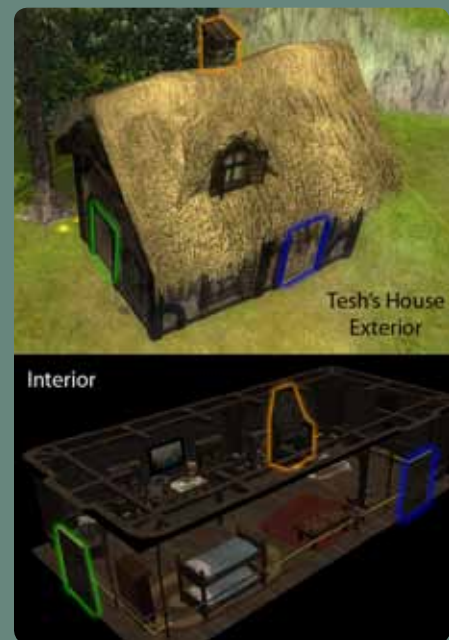
The mentioned scripts from the previous section don't cover everything, but were just a few samples of problems or needs I had, and how I used scripting to overcome them. As I mentioned, prior to this, I really didn't have much of any experience with coding, unless I was poking around in blog layout templates, cutting and pasting sections to move them around, or manipulating things like border sizes and colors. I'm not afraid to poke around under the hood, though, and pick up quick on key concepts. I can *look at code* and understand what it's doing. Once I grasp the initial concept, I'm able to think abstractly and come up with new solutions.

One of the more frustrating parts of working with the NWN scripting toolset was the sheer amount of false leads you come across. The engine has all sorts of loose ends just dangling out in the breeze. Many of the functions you can come across are legacy leftovers from NWN1, or were never fully implemented. They're there, but they simply don't work.

NPCs don't sit down, for example. There's a function in the script editor you can throw into scripts called `ActionSit`, and compiling the script gives no errors that anything is amiss, but then it simply won't work in the game. Digging around on wikis and toolset forums, you come to realize that the devs just didn't really bother making it work. It was troublesome, so they just... let it be. Getting Melf and Crimson to lay down on the floor of the sickhut was a big operation unto itself using custom animations on that area's '*on client enter*' script. People have gone to great lengths to create custom mods with 'usable' chairs that you can use as a Player Character for RP purposes, and you can brute force an NPC to sit in a chair if you jump through a bunch of hoops, but bumping into them as you walk past them upsets this delicate state, or if you assign ambient animations to them, they'll spin in the chair (right through the chair back!) to face you as you walk by. It's sloppy.

Once you realize this, you begin to notice how all the NPCs in other people's modules are just... standing around... in bar areas. It's kooky. Another thing that made no sense was a complete lack of windows in interior tilesets. I specifically went to great lengths to try and lay out my interior areas to match what the building looks like from the outside. If there's a chimney on the back edge of the house, and doors on the facing and left walls, I put those in the same locations internally. This seemed like... duh, right?

But then, looking in the toolset tiles, there are no walls with windows, and no window placeables to stick on the walls. It didn't make any sense, and again... I began to realize that unless you were using the "super intricate castle" tileset, you were just out of luck. It wasn't my module only, it was an entire toolset feature! Everyone's bars are these dark, cave-like areas. They drop sunbeams shining down from the ceiling to lighten the place up, but... the sun is just shining in through the ceiling beams, I guess?



Little things like these don't break the game, they just seemed like odd little quirks, and I didn't lose sleep over them. Using the Neverwinter toolset, I stayed within what it offered, while silently noting things like this. The Dragon Age toolset, I noticed, has windows galore, and all kinds of neat things you can do with light streaming in through them (more on that in a sec).

One last detail I want to mention 'not losing sleep over' is the combat system. I never really *played* Neverwinter Nights when it was released, and my only real exposure to the end user side has been poking around in other people's modules, and running the included campaign for about an hour. The combat system in Neverwinter Nights is... very... ummm... I'm trying to think of a good way to put this...

I saw the Chinese Circus once when they passed through my area in Japan. We got the whole family together and went and saw a show. A good time was had by all. There wasn't half as much *dodging* and *tumbling* that day at the circus as I see when I'm killing a rat in NWN. So many roll checks! Roll checks for initiative! Roll checks for opportunity! Roll checks to see if my roll checks are beating your roll checks! I'm hungry, let's have a roll check to see if we should go get lunch! Tacos or hamburgers? Let's have a roll check! I *get* that they're trying to stay true to the whole D&D 1d20 thing, but I found myself losing roll checks to stay awake, so I ended up just giving our protagonist a mace with ridiculous chance to hit and +damage done against undead (one of the two mob types we encounter, the other being your basic rat).

I've been spoiled by WoW, where pushing a button produces immediate results on my screen. Not only is the action's animation immediate, but there's usually a sound notification of whether your attack landed or not, and visual feedback that something happened. Whether I'm thunder-clapping on my tank, healing on my priest, or immolating someone on my warlock, there's an immediate feedback that *something is happening* that I simply could not find in NWN.

I've written long posts on how important combat is in a game, since it's what you're doing 95% of the time. This isn't the place to go into a huge discussion about combat systems (but I'll happily go into it later if desired!). Just suffice it to say that I wasn't losing sleep over how boring I felt NWN's combat system is. I was telling a story, and the story involved our protagonist discovering undead wandering around at one point. The fact that she's able to three-shot them and go along on her merry way doesn't make or break the flow of the narrative. I even included in the dialogue text that she had no issue with them, difficulty-wise, but that they were *there*, and that someone should probably look into that.

So I sped up the fights, and even included a stack of health pots in your bags should you need them. The main point of combat is that there is some, and it's triggered in various ways, not that it was some amazing struggle. That wasn't the point of my module, so I steered around it. I have plenty of interest in in-game combat, but this story, combined with this fight system, just wasn't the avenue to pursue it.



Spoiler alert

A quick rundown on how to achieve each of the ten endings. For the lazy cheaters who only want to see the end movies, just go into the My Documents\Neverwinter Night 2\Movies folder and grab a bag of popcorn (and the Bink Player from RADTools).

Otherwise:

- 1) **The Good Ending** - Help Hatch cure the virus, and just follow the initial quest chain to its full conclusion. Make sure to *peacefully* retrieve the missing journal page from Tesh through dialogue in his hut during the quest There and Back Again.
- 2) **The Almost Good Ending** - Help Hatch cure the virus, but at the last second just tell Tesh to shut up with his rambling and hand over the page. He'll attack, and you'll loot the page from his corpse.
- 3) **Odd Jobs** - Ignore the virus, and just do odd jobs in town to scrape together 100g. Handling Larisa's bar tabs, Jong's rat problem in the orchard (and selling the gem you receive), and fetching Ore for Syp will get you enough gold to purchase a boat ride in Pinole.
- 4) **Help Syp** - Syp suspects that Gevlon is stealing from him, and asks for you to acquire a ledger from Gevlon. Whether you pickpocket it from Gevlon, or kill him outright is up to you, but both choices lead to this ending.
- 5) **Blunt Trauma** - Through a conversation choice with Tobold, you can opt to just kill the sick in town, thereby eliminating the problem of the virus at the source. Use the mace you carry in your bags, and just bash their faces in! Rawr!
- 6) **Mercy Killing** - Same as #5, but insist on doing it in a dignified manner. Tobold will direct you to Spinks, who will supply you with a dagger named Quiet Death that will ease the suffering of those you're about to put down.
- 7) **Kill Tobold** - Just whack Tobold, loot his second floor key, and grab a voucher for travel out of his upstairs safe.
- 8) **Kill Syp (Help Gevlon)** - Gevlon 'wants to make Syp disappear'. Do the dirty work, and collect your reward.
- 9) **Kill Gevlon** - Just whack Gevlon, loot the key to his shed, and make off with your ill gotten goods.
- 10) **Blackmail Gevlon** - Pickpocket Gevlon's ledger (you'll need to talk to Syp, and basically start quest #4, above), but then, rather than return to Syp with it, inform Gevlon that you have his book, and what you intend to do with it. He'll pay you off to just leave town at once.