

Project Number: DMO - MQP 3378

Veniss Underground: A Neverwinter Nights
Modification

A Major Qualifying Project Report
submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
By

Aaron Bell

Joshua Brunelle

Garon Clements

Date: April 24, 2008

Professor Dean M. O'Donnell, Major Advisor

This report represents the work of three WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review.

Abstract.....	3
Introduction.....	4
Adapting “Veniss Underground” to the Game World	5
The History of Veniss	5
Creating the Quest Story.....	8
Writing the Dialog.....	9
Visualizing Veniss	11
The Artistic Workflow	11
Bringing Veniss to Life.....	15
Scripting and Assembly	15
Conclusion	17
Bibliography	18
Appendix 1: Design Documentation and Resources	19
Section 1: “Veniss Underground” Outline	19
Section 2: “Veniss Above” Design Document	25
Section 3: “Veniss Below” Design Document	33
Appendix 2: Workflow Tutorial.....	39
Maya to NWN.....	39
BUILD.....	0
POLYTRANS/GMAX	1
3DS	4
ALMOST THERE	10
Appendix 3: Sample Scripting	15

Abstract

Veniss Underground is an Adventure RPG hybrid set in a futuristic, dystopian city named Veniss. The game is divided into two separate acts, each with its own unique protagonist. The storyline of the game is loosely based on the book “*Veniss Underground*”, by Jeff Vandermeer. The game is primarily composed of adventure-style exploration and d20 style combat.

We created the game a *Neverwinter Nights* module, which allowed us to save valuable time by utilizing the existing toolset and graphics engine. Garon was in charge of all of the scripting and most of the more technical aspects of the toolset. Josh developed the necessary artistic workflow and imported the art assets into the toolset. Aaron and Josh worked on level design and texturing, which included the creation of tilesets, placeable art assets, and character models. Aaron was also in charge of adapting the story, creating the design documentation, and writing the dialog and cutscenes for the game.

Introduction

The goal of our project was to create a module for the preexisting game *Neverwinter Nights*, utilizing the developer-released toolset. Besides this initial prerequisite, the use of *Neverwinter Nights*, we were allowed to decide all other facets of the project. Because of the many completed NWN modules and modifications we decided early on that our project would try to use the toolset and engine in a different and more creative way than had been done previously. At the same time we did not want to get bogged down in writing out an entire world setting, back story, and storyline for the project. To satisfy both of these concerns we chose to adapt a preexisting setting to our project that would be unique in its own right and especially different from the traditional “sword and sorcery” setting. After a brief search we chose Jeff Vandermeer’s novel *Veniss Underground* for our setting and much of our storyline. We contacted Jeff and he was very helpful and excited about the project. He gave us full authority to use and modify the story and setting in any way that we deemed necessary.

After we had our setting we began designing the game. We planned out extensive design documentation, detailing the overall story arc, its segmentation into two distinct acts, the unique protagonist assigned to each act, level by level break down, and various notes on the back story, Veniss, and the characters. Because of the heavy art orientation of the project (two art students and one technical student) we planned on creating the world from the ground up, including the fabrication of 3D tilesets, placeable models and textures for every level and many characters. Discoveries made later in the project would make this early decision much more ambitious than originally envisioned.

Adapting “*Veniss Underground*” to the Game World

The History of Veniss



An alley in Veniss

The history of Veniss is a long and winding one. Ten years prior to the time of the book “social planners” ruled the city. I believe that the social planners were a form of authoritarian artificial intelligence. Quin helped the government during this time, running an artificial birth program, vat-growing human beings. During this time the city was known as Dayton Central. The government collapsed after a violent insurrection.

The artificial intelligence(s) were defeated and the city was renamed Veniss. The “social revolution” began, bringing about a new appreciation of new, “living art”, free trade, and the death of

all forms of “old art”. With free trade came the privatization of the police, large amounts of waste, and pervasive advertising.

Veniss, like all cities in the book, is a self-contained city state, walled off from the rest of the world. Human wastefulness has destroyed the world outside the insulating wall of Veniss (and other city states) and all that remains is a chemical desert. Veniss, which originally had active relations with Balthazar, the nearest other city state, has increasingly become detached from the rest of the world.

A once flourishing off-world mining colony operation has faltered to the point of only barely providing subsistence to the colonies themselves. The Canal District, an area of Veniss full of ocean canals that spaceships bringing goods from off-world used for “wet-landings” is entirely unused by the colonies now.

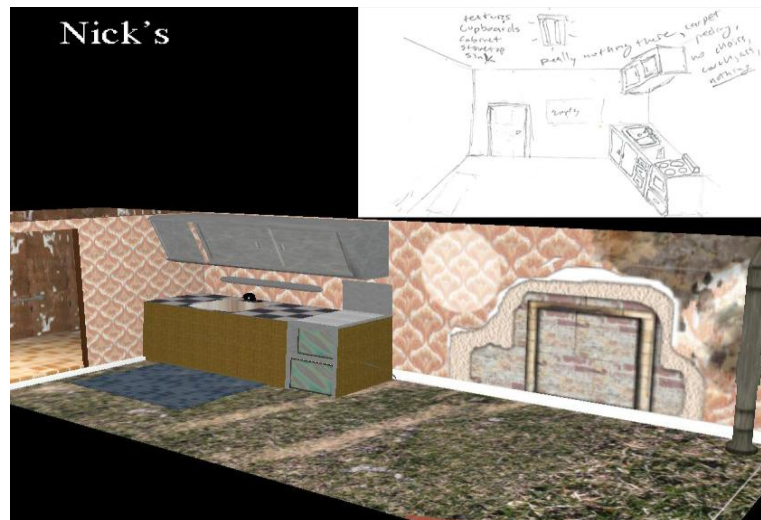
The city itself is slowly collapsing, moving always further towards barbarism and chaos. “The Bastion”, where Nicola and rest of the city’s programmers work, is the epicenter of the only real concerted effort to keep the city running. Programmers at the Bastion work to keep the trains and other automated social services in continued operation. They use dated technology that still employs the artificial intelligence that once dominated the city, but it is “carefully yoked for obedience.”

Much of this state of affairs is due to the fractured and disorganized nature of the city’s government. The city is ruled by many disassociated pockets of authority. For instance, when Nicola travels to Nick’s apartment we learn that “a trip from one end of the city to the other requires eighteen security stops.” Furthermore, this governmental disorder, and the revolution that led to it, has caused a complete break between Veniss Above and Veniss Below. Veniss Below, originally created to mine deep into the Earth’s surface in order to bring up raw resources to the surface, has

become lawless and disconnected to the surface. The mining operations that continue have no relation to the surface and people literally live out their entire lives in the hell of the underground for no reason whatsoever. A long standing lottery system, which allows a certain number of individuals to come to Veniss Above every year, continues to operate but individuals who do win the lottery have no way of telling their families the truth of the situation once they have learned it.

And throughout all of this, Quin and his “Shanghai Circus” cast a long, black shadow across the city. His control above ground is powerful but limited, mostly supported by his continued patronage of the city through gifts of his “living art” (meerkats and ganeshas). However, below ground Quin is practically a god, controlling nearly all levels absolutely.

Creating the Quest Story



Nick's Apartment with Concept Art

In order to adapt Vandermeer's sprawling story we outlined the entire novel, fully summarized the parts that we felt were essential to the story and the game world, and then modified the story where necessary.¹

Directly after outlining the full story we decided that it would be too long and that cuts would be necessary. We rewrote the ending of the story both to fit time constraints and to avoid some of the more extravagant settings that would have been impossible to create within the NWN engine and timeline.

After shortening the story we went through and broke it down into areas to be explored, the assets necessary for each area, and the quest information to be gained. Following a template design document provided by our advisor we wrote two major documents, one detailing each of the two acts

¹ See Appendix 1 – Design Documentation, *Veniss Above* and *Veniss Below* for outlines and summaries.

in the game. Each document (which can be reviewed in Appendices 1 and 2) are separated into history, goals, quest story, and level design. These documents helped us maintain a consistent vision for the project without confusion.

Writing the Dialog

When writing the dialog for the project we pulled heavily from the existing dialog and other miscellaneous passages throughout the book. Vandermeer's writing was quite good and the game was designed to be heavy on dialog so we chose to use many passages directly from the book. We also, of course, modified existing writing and wrote much of it ourselves. We often altered or added to the existing text to either make the dialog choices in the game more interesting and dynamic or to include interesting material from the novel that otherwise would not be included in the game.

During the course of writing the narrative and dialog for the game we experienced firsthand the major differences between writing for a traditional story and an interactive narrative. While adapting the novel into a more interactive narrative without losing what was special about the story was often challenging, two key features of the novel made this process much easier than it would have been with most novels.

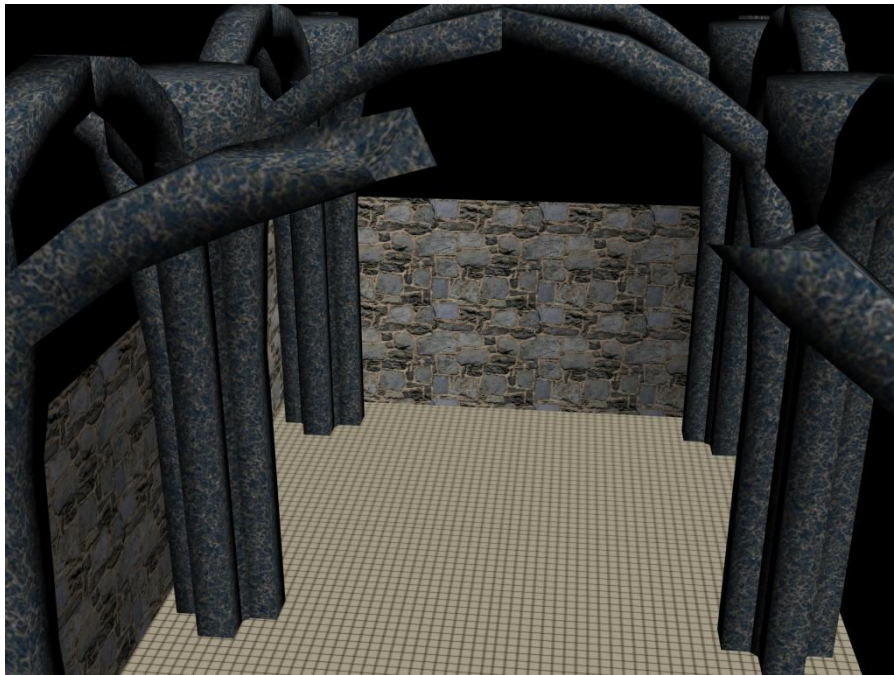
First, the story was already divided into three distinct acts, two of which we would eventually choose to include in the storyline of the game. This division of the story helped us to follow a more traditionally segmented interactive narrative outline, with cleanly separated acts and levels which roughly followed the flow of the story from chapter to chapter.

Second, the perspective of the novel was often very helpful. The second act of the book is entirely written in a second person perspective, which is essentially as close as traditional prose can

get to interactive narrative. In many adventure and role-playing games the character is often told what they are doing, specifically in cut scenes and other key points that progress the narrative in an important way, and subsequently many passages required no adaptation to use directly in the game. The third act of the book, featuring the Shadrach character and marking the beginning of the second and final act of the game, is written in the third person perspective. While this was not as helpful as the second person perspective employed in the second act of the novel, it was much easier to work with than a first person perspective (which was how the first act of the novel was written).

Visualizing Veniss

The Artistic Workflow



The Cathedral Area

When designing the artistic feel for the project we were initially very confident that modifying the artistic assets of the game would be relatively easy and would give us a significant amount of freedom to create a look all our own. *Neverwinter Nights* is famous for being easy to modify and this initial impression was one of the major reasons why the game was chosen as the basis for an IMGD project. Additionally, NWN is well known for its extensive and passionate mod community. The project progressed under the assumption that while an alternative setting may be challenging, it was entirely possible given the ease of modifying the game and the helpful and well established mod community surrounding it. It was not until after we had confirmed the use of the “*Veniss Underground*” as our setting and nearly completed that design documentation for the project

that we first had the creeping suspicion that something might be wrong with the importation of custom art assets. This was quickly confirmed and thus began weeks of mostly fruitless experimentation in ways to forcefully push our assets into the *Neverwinter Nights* engine.

After many hours of experimenting with multiple programs and tools, a work flow was discovered between Maya 8.5, our main modeling program, and the *Neverwinter Nights* engine. Maya was first used to model our areas, as it was the program we were most proficient with, and this quickly presented itself as the first problem in our workflow. There was no definitive way to translate our models from Maya to a format readable by NWN. After reading through tutorials, it seemed that we would need to use 3D Studio Max 5 (3DS) in combination with a script set, NWMax, and the MDL suite plugin set. At this point, there was still the initial problem of getting our models out of Maya, and into 3DS without missing faces and other problems that occurred when using the 3ds exportation plugin for Maya. That is when the program PolyTrans was discovered to convert Maya's binary format to 3DS's 3ds format. PolyTrans, while much more successful than any prior exportation techniques, was not perfect. We also discovered that slicing geometry for use as tiles within 3DS 5 was not possible, so GMax, a program designed exclusively for game asset production, was found for this purpose. With these basic programs for handling the geometry collected and a workflow beginning to solidify, it was time to get something into the game.

From Maya, a third party plugin was used to export the geometry as an object file. PolyTrans was able to read this format and allowed us to efficiently convert from Maya binary to 3DS Max format. 3DS format is readable by Gmax, allowing us to use NWMax for the GMax script set, which allowed us to slice and export the geometry into tiles, which is a prerequisite for importation into *Neverwinter Nights'* native .mdl format. This completed the creation of the tile but it was not the end of the process. The tiles were invisible to the toolset at this point and useless without a walkmesh. Furthermore, the textures at one point were also being severely distorted by tile slicing, but we

eventually learned to simply re-texture in 3DS as a work around. This was necessary because the slicing would destroy the UV coordinate information originally used to map the models, so this information had to be recreated in 3DS 5.

The walkmesh is basically a collision map, meaning that anywhere wherever it covers the player cannot walk through. The mesh works in two dimensions, with height being left out. Without a proper walkmesh, the player is launched above the tile, making it unusable. This proved difficult to diagnose and repair, but we finally found out that it needed to be built in 3DS, not Maya. This forced us to learn how to use 3DS efficiently, if only to build planes and manipulate vertices. In order to construct the walkmesh, we imported the tiles into 3DS, and learned how to texture using 3DS. Once we retextured, it was time to export the tiles once again.

Once exported, the *Neverwinter Nights* Toolset (NWNT) needed a way to see them. Luckily there were thorough tutorials on how to do this², and with a few more specialized tools it was accomplished. The .bif unpacker exposed the game files for modification, and the .set editor allowed for addition of our custom tiles to an existing tileset. Once in the tileset, any minute problems would expose themselves, and constant editing, restarting NWNT, and testing again eventually worked out the kinks.

The time it took to discover workflow should not be underestimated. There were many, many unsuccessful combinations of tools, scripts, and programs that would get very close to having something working, but would finally fail. Many iterations of the workflow proved nearly successful only to fail at a step that a previous flow had succeeded at. One combination yielded walkable tiles,

² Bioware Corp. "Tileset Construction." [Obtained from Bioware's own website as a free download to registered users.]

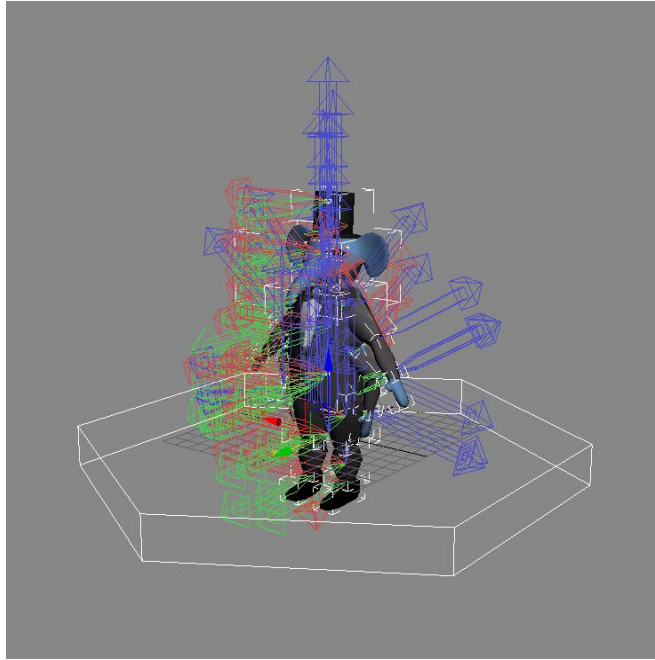
but destroyed all textures at the same time. The workflow diagram seems very easy to follow, but if you insert the inevitable problems that arise along the way, the graph gets much more complicated. For an idea on how much work was put into getting a simple two tile area into the game, see the tutorial in Appendix B. It took a long time to discover something this complicated, but fortunately it did produce reasonably consistent results. All this work has resulted in a manageable but not exactly easy workflow. It is easy to follow once completely documented, but still quite a tedious process.

The creation of placeable objects was much easier, though it was not without its initial problems. An existing placeable was imported into 3DS to get the base we needed. Then, after translating our objects made in Maya, we imported them into the scene in 3DS. After re-texturing and parenting our new objects correctly, we exported them under a different name. From there a text file was modified to accommodate for our new object file. In the toolset we placed any random existing placeable and then changed its appearance property to our new item. The same goes for NPC models, but assembling the geometry for those was much more difficult. Every part of the NPC model had to be linked to a hierarchy of parts. Additionally, hands, feet, shins, thighs, forearms, biceps, head, neck and torso pieces all had to be textured separately. Each piece also needed to have its pivot manipulated in order for *Neverwinter Nights'* animations to work correctly. This seemed like an endless cycle of changing some geometry in 3DS, launching the toolset, testing it in game, closing the toolset, and doing more work in 3DS. Eventually the models started to appear without huge problems.

After extensive research of the existing modifications and modules created for NWN we have come to realize that nearly all of them involve only slight alterations of existing art assets. Re-skinning creatures, NPCs, tiles, and even items seems to be the norm. Making brand new assets from scratch and using very little of what existed already proved to be a serious challenge for the project. Almost everything in the game is custom made and textured. Thanks to the workflow we eventually

came up with, everything got into the game working, despite the huge obstacles that we initially faced.

Bringing Veniss to Life



Rigging and Importing a Ganesha

Scripting and Assembly

Neverwinter Nights is famous for its easy to use and intuitive scripting and assembly process. Thankfully, this notoriety was well deserved and the majority of the scripting was easy to do and was accomplished quickly.

The scripting was all done using the scripting language included with the *Neverwinter Nights* Aurora Toolset. The scripting language is very robust and is used to do just about everything in the engine. The language was efficient and easy to use, and subsequently most of the scripts written for this project were small in size. The scripting language has many predefined functions which allowed

us to accomplish most of the necessary scripting goals quickly and efficiently. Each placeable object, trigger, and waypoint is given its own unique tag when created and these tags are used to reference certain objects within a script, making it intuitive and efficient.

We found a cutscene engine on the *Neverwinter Nights Vault*³ written in the *Neverwinter Nights* scripting language and by utilizing this we were able to create the several cutscenes found in the game. The cutscene engine was so useful that all the games cutscenes were created by simply using the functions provided by the cutscene engine.

The toolset itself was very easy to use to assemble levels. In normal, non custom levels it is very easy to place objects where you want them and to set up all the triggers necessary. When it comes to custom content though, there are many things that need to be corrected when the levels are created in order for them to work in the toolset. One of the major problems with the custom content is the walk meshes. Any problems with the walk mesh it made it hard to place objects exactly where we wanted them.

Another problem we ran into, as a consequence of the custom art assets, was the implementation of transitions from area to area. We were not able to use doors in the way *Neverwinter Nights* normally would. In order to get around the area transition issue we were able to use waypoints and triggers, or scripts on objects to simulate how doors function in modules with non custom levels.

We established a pattern for assembling levels by creating an area and then placing the custom level within the area. We then placed the placeable objects in the area and set up all the waypoints and triggers. Scripts were then written or reused and set to each object or trigger where

³ www.nwvauld.ign.com

necessary. After all objects and scripts were placed the level is tested to make sure everything works as intended.

Conclusion

Despite a few problems in art creation, the project as a whole was mostly a success. Once we developed a consistent workflow we were able to complete most of the game that we had ideally envisioned and we documented the valuable lessons we learned so that future groups at WPI can benefit from our work.

The biggest thing we learned was how vital it is to be familiar with the toolset and engine before laying out the project and writing the design documentation. Luckily this won't be nearly as much of a problem for any subsequent groups because of their foreknowledge of the strengths and weaknesses of *Neverwinter Nights* and our tutorials, if they do choose to pursue an alternative setting. We recommend that any groups in the future focus on the toolsets strong points, including dialog creation, scripting, and environment creation. All of the areas are well developed and intuitive and should most likely serve as a foundation for any future projects. While it is possible to force the toolset to do things that it wasn't really intended to do, we recommend that most of the project be planned around working with what it was designed to do, as it does those things very well.

Overall, the final project came close to our initial expectations and will serve as a foundation for future work, so we consider it a success. In the end, our biggest problem was time, not the engine or toolset, so we are optimistic that future groups will be even more successful. Knowing what we know now, any group that can avoid or plan for the problems that we encountered should have more than enough time to complete almost any time of project they can imagine using *Neverwinter Nights*.

Bibliography

Bye, John 'Gestalt'. "Gestalt Cutscene Scripting System".

<http://nvwvault.ign.com/View.php?view=Scripts.Detail&id=2570#Files>

Ratt, Samurai. "Energy Pistol & Rifle Beta AB".

<http://nvwvault.ign.com/View.php?view=Hakpaks.Detail&id=6015#Files>

Ahearn, Luke. *3D Game Textures: Create Professional Game Art Using Photoshop*. Canada: Elsevier inc., 2006.

Mr. X . Tutorial: Creating Custom Tiles, NWvault,

http://nvwvault.ign.com/?dir=resources/tutorials/&content=tile_tutorial

Bioware Corp. "Tileset Construction." Copyright 2002, Infogrames Entertainment. New York, NY.

[Obtained from Bioware's own website as a free download to registered users.]

Christopher. Bioware custom content forum post. Tuesday, 26 July 2005 03:06PM.

<http://nwn.bioware.com/forums/viewtopic.html?topic=162084&forum=48&sp=45>

Wikipedia. "Texture Synthesis". http://en.wikipedia.org/wiki/Texture_synthesis

Appendix 1: Design Documentation and Resources

Section 1: “*Veniss Underground*” Outline

Plot Summary

ACT I

[Nick’s Perspective]

- Nicholas has recently been burglarized by “pick dicks” and asks Shadrach for information about Quin so that he can go ask for a meerkat for protection
- Shadrach initially won’t tell him anything, saying that Nick will get hurt if he goes to Quin, but then gives in, telling Nick to tell Quin that he was sent by Shadrach and to *only* ask for a meerkat.
- Nick goes to Quin’s, which is hidden behind a false, holographic wall at the end of an alley.
- Nick asks Quin (not the real Quin, just a puppet) for a meerkat and then, when he doesn’t get the answer he wants, offers to work for Quin.
- Quin’s office is filled with caged creatures and when Nick offers to work for Quin the cages open and the animals hold Nick down and inject him with a sedative.

ACT II

[Nicola’s Perspective]

- It has been two weeks since Nick has missed an “infrequent lunch date” and Nicola is worried about him
- An additional three weeks pass without any word from Nick.
- We learn that Nicola works as a coder for the “Bastion”, an institution that literally keeps the trains running on time. They stand between Veniss and a collapse into barbarism and chaos.
- Nicola uses her position at the Bastion to search for Nick. She finds that: he hasn’t paid rent for two months; he is behind on all of his bills, he has several dubious loans out, all of his credit is maxed out, and that his last purchase was made five months ago.
- She hacks into his bank account and discovers that a new bank card was issued to him less than three weeks ago and that a recent purchase was made on it from the 10th level underground (a level that she didn’t even know existed).

- This is worrying because he could have been killed or may be hiding from outstanding loans from dangerous people.
- This is also a cause of guilt because the last time Nicolas saw Nick he asked her for money and she wouldn't give him any.
- She visits Nick's apartment in the Tolstoi District, an arduous journey through six other districts of Veniss and a plethora of security check points.
- The district is very creepy with many deformed animals hiding in the shadows of the buildings.
- She gains access to his apartment and everything, including his inexpensive furniture, is gone. The apartment smells of animal fur.
- Upon further inspection Nicola finds animal hairs in the kitchen and, inexplicably, the kitchen is bare of anything. It is *too* clean. Something had to have happened there and then been cleaned up after.
- She finds a piece of paper with a poem about Quin with what appears to be blood stains on the lower left corner.
- She leaves the apartment only to find one of Nick's hologram art pieces in the hallway. It wasn't there before. She smells animal musk and leaves as soon as possible knowing that she has to find Shadrach.
- She finds Shadrach in the Canal District, where he always is.
- He lies, saying he hasn't seen Nick in three weeks. He is cryptic but on the verge of telling her something when a meerkat appears from the crowd and leaves with Shadrach. It is unclear who is leading who.
- Nicola wakes up to someone knocking on her door. It is a ganesha and a meerkat. The ganesha is dressed in a tuxedo and delivering the meerkat, a gift from Quin, because "a friend of Shadrach's is a friend of Quin's" (60).
- She is wary of accepting the gift but in the end feels that she has no choice.
- She names her new meerkat Salvador, "after that grandmaster of the Dead Arts and godfather of the Living Arts" (61).
- Nicola questions Salvador about Quin and he is, at first, defensive about the subject of his creator.
- He then proceeds to recite a few choice lines from the poem about Quin that Nicola found in Nick's apartment but says he knows nothing else. This is very suspicious/ominous because there is no way that Salvador should know about Nick's poem.
- Salvador proceeds to clean up the apartment and Nicola leaves to go shopping.
- She purchases a brand new laser gun for protection.
- She learns that Salvador has a mysterious way of getting fiddler crabs.
- The next day she asks about Quin again and Salvador gives her more, but generic information that he "looked up at the public archives". This is very suspicious because a. they allowed an animal to access the archives and b. Salvador knew to do so in the first place.
- A day passes. No information can be found and Shadrach can't be contacted.
- Nicola wakes up at 3 am and witnesses Salvador sneaking back into the apartment with more fiddler crabs.
- She questions him and he explains that he went out early to get the best ones but his story is very suspicious.
- The next night she waits until she hears him leave and follows him into the city.

- After almost losing him several times she follows him into what appears to be a dead end alley.
- She attempts to lean against this wall and falls through it; it is only a holographic image of a wall.
- She falls into a perfectly preserved piece of wilderness, somehow hidden between the skyscrapers and other buildings.
- She follows two created animals down a white stone path and across a bridge (over the mudflats where Salvador captured the fiddler crabs he brought home at night) to another wooded area with a mysterious light.
- She follows the light and finds a congregation of meerkats and ganeshaas playing and watching a holographic film. The film is a compilation of horrible acts of violence committed by humans towards animals. The film continues to show humans killing other humans, which more visibly shocks the animals than anything else.
- She panics and runs away, with the phrase “*you are not superior, you are not superior*” repeating in her head over and over again (80). She realizes that Quin’s creations are destined to wipe out humanity.
- Nicola waits until Salvador is home, ready to kill him if necessary. Salvador smells where she has been and knows what she has seen before she can say anything.
- Salvador tries to kill her but only manages to bite her wrist before she shoots him twice, incapacitating him (84).
- Directly after this Nick shows up and strangles her.

ACT III

[Shadrach’s Perspective]

- Shadrach, following orders from Quin, is visiting “Lady” Ellington.
- He notices that she, in contrast to her old withered body, has a new left ear, left eye, and left hand.
- He notices that her left hand has a remarkably similar birthmark to one on Nicola’s left hand before he realizes that all the new body parts are from Nicola.
- He realizes that Quin has ordered him to visit the Lady so that he could make that exact connection.
- He goes to his apartment and grabs his older model laser pistol and his badge issued to him by Quin.
- He travels to Quin’s office and gets the “donor” number of the body parts that the Lady has recently acquired.
- He finds that Nicola, if she is still alive, is on the fifth level underground, “Slade Organ Bank”.
- She is listed in “live storage”, so there may be a chance that she is still alive.
- He prints out Nicola’s organ donor number and as he is leaving Quin’s puppet maliciously asks if he has visited Lady Ellington’s.
- He heads to Nicola’s apartment.

- He finds scuff marks from where her couch has been moved, Nicola's laser gun, two red stains that appear to be blood and rotting fiddler crabs in the sink. He looks in her closet and finds Salvador, still alive but dieing.
- It is obvious to Shadrach that Salvador is an assassin subtype and that he was sent to Nicola with the goal of killing her.
- He incinerates all but Salvador's head (knowing the assassin subtype can live for a long period of time without its body) and glues it to a plate.
- He names Salvador John the Baptist, pulls out John's sharp teeth, and puts him in his large trench coat pocket.
- He eats a last good meal at a café and decides that he has to save Nicola.
- He enters through a manhole in an alleyway, allowing him to bypass the first two levels.
- He is attacked by four little creatures but manages to fight them off.
- They leave a bomb next to him and jump off the edge of the terrace they are fighting on, deploying parachutes.
- He drops the elevator they were fighting on and it crushes them.
- He places the dormant bomb in John's ear.
- He continues his descent, encountering local sublevel dwellers and various forms of law and authority.
- He reaches the fifth level. There is a 4 mile long line of people waiting to either "donate" or buy from the organ bank.
- He pushes through, brandishing his gun and the badge from Quin (109).
- He enters the antechamber of the organ bank and is allowed to pass once he tells them he is on business from Quin.
- He enters into the cathedral like bank and meets Dr. Ferguson, a surgeon that works their.
- Shadrach gives Ferguson the donor number printout Ferguson confirms that she should still be around and possibly alive.
- Ferguson leads him to a "writhing, seething pit of flesh" (117) but leaves him after this.
- He enters the pit and the attendant guides him to the back where the fresh bodies are kept.
- He is guided to another room where a naked dwarf oversees a giant pile of legs and other body parts. The dwarf indicates that Shadrach must search in the pile for Nicola.
- After a half an hour of swimming through the pile of body parts he finds Nicola at the bottom, still hooked up to life support.
- He takes Nicola and escapes the cathedral and continues running until he finds his parents old house.
- They are gone and a creation of Quin's named Candle lives there now. He is a priest agrees to help Shadrach find a psychewitch to help Nicola.
- Candle takes them to Rafter, a psychewitch, who agrees to help them, but the process will take three to four days.
- Rafter, who can use a machine to read people's thoughts, finds that someone has recently accessed Nicola's. This allows Shadrach to view Nicola's thoughts without hurting her, which he does.
- Shadrach learns of Nick's betrayal and that Nicola definitely doesn't love him.
- Shadrach decides to travel to the tenth level to kill Nick while Nicola is being treated.
- He talks to John who tells him that he only has twenty four hours left to live and that attacking Quin is suicide for Shadrach.

- Shadrach enters the tenth level, the garbage zone. “The garbage zone was a revolving beast that ate its own dark trail ... once it had been AI, but now it was just an old beast, and a slow beast” (140).
- Shadrach sees the garbage folk (people who live their entire lives just harvesting the usable waste that the citizens of Veniss throw away).
- After some exploring he finds the garbage folk stabbing some creature and he scares them off. The creature is Nick.
- Nick has been transformed into a disgusting, failed creation that he made when he was a child.
- Nick explains what happened between him and Quin. He explains the evils he has committed, specifically how he was instructed by Quin to murder Nicola.
- Nick explains Quin’s plan to free the meerkats so that they can be free to replace humanity.
- Nick accidentally gives away that Quin is on the thirteenth level and Shadrach forces him to take him there.
- After using a series elevators, escalators, etc. they arrive in an ancient subway station. They wait with strange, phantom passengers for the train to arrive.
- The train itself is full of holes and travels at amazingly fast and alarming speeds.
- They purchase parachutes because they have to jump out of the train over a giant hole that leads down to Quin’s kingdom.
- Nick attempts to betray Shadrach by trying to push him off the edge and sever his parachute.
- Shadrach manages to open his parachute after falling out of the train but is hit by two people committing suicide as they fall to their deaths.
- He hits the ground hard but is not seriously wounded.
- Shadrach awakes to find that he is on the shore of a vast ocean. He realizes, as his eyes adjust, that Nick and the other suicides are dead next to him.
- The thirtieth level is hell on Earth; “a thousand lost souls populated the land along the shore, condemned to wander until death” (162). This place is where Quin’s failures and mistakes are sent to die.
- John and Shadrach talk more and while John won’t help Shadrach find Quin he recommends that he use a luminescent slug that also happens to be a map to find his way.
- Shadrach touches it and it unfolds outward, the most beautiful thing he has ever seen. It forms a three dimensional map and when he touches Quin’s headquarters it begins to flow, like a bird, towards it.
- While following the map he meets the Gollux. The Gollux tells him that he can’t trust the map but John becomes furious and says that the Gollux is lying.
- The map dies and the Gollux shows Shadrach the correct way to Quin’s – across the ocean.
- The Gollux summons a saylber, a manta ray like creature to transport them across the ocean, to Quin’s.
- On the way they see vast numbers of meerkats fleeing from Quin’s city in floating wooden cities.
- When they arrive Shadrach finds that Quin’s city and home are constructed inside of a giant, nerveless fish’s mouth. The fish can feel no pain, allowing intricate cities to be carved out of its mouth flesh and it swims in place forever, even as it rots apart.

- As they get closer they can see all of Quin's creations are killing each other on the surface of the great fish.
- They travel into a replica of the forest Nicola found in Veniss. On the way they see creatures of Candle's breed, who serve as priests for Quin, crucified because Quin no longer wants priests
- They find Quin, who is essentially a puddle of human flesh who's only form of communication is by way of another puppet, similar to the one Nicolas first encountered.
- Quin explains his plans, divulges that he may have even created Nicola and Nicholas during his days working for the government.
- Quin forces the Gollux to try to kill Shadrach but before he can Shadrach kills him.
- Quin explains that meerkats are destined to replace humanity and that he designed them to be this way.
- Quin presents him with a moral dilemma: Quin wants to be killed so that he can become a martyr, which will drive his creations to overthrow their human masters BUT if Shadrach does not kill him he will have Nicola killed. So Shadrach must choose between quickening the destruction of humanity or saving Nicola's life.
- Shadrach chooses to destroy Quin and blow up his lab with bomb in John's ear. John is destroyed by this.
- He flees Quin's fish, almost drowns in the ocean that it swims in but makes it to shore.
- He rests, then climbs the rock face all the way back to where the train tracks lead to the subway station.
- He pushes through all his pain and exhaustion and makes it back to Rafter's, where he sees that Nicola is recovering.
- Shadrach pays Rafter and he and Nicola head for the surface, for a hospital.
- They make it through without significant challenge.
- Shadrach finally collapse from exhaustion, having not slept for a week, and the book ends, not telling us if Nicola has come to love him or if the meerkats will revolt or anything like that.

Veniss Underground

Veniss Above

9.13.07

1. Latest Changes

- [9.13.07] Document Created
- [9.14.07] Sections 5.2, 5.3, 6, and 7 created
- [9.22.07] Latest Changes section updated
- [9.22.07] "Veniss Above" document renamed "Veniss Above" (instead of "Veniss_Above_9.13.07")
- [9.23.07] "History of Veniss" and "Quest Story" sections updated
- [9.27.07] Added notes to section 5.2 "Nick's Apartment" on fight scene

2. Goals

- Create an interesting, aesthetically pleasing futuristic city
- The level begins with Veniss_Cubicle and concludes with Veniss_NicoApt
- Introduce Nicola, begin and advance the story through first major quest, “Find out what happened to Nick.”¹

- Get the player excited about the story
- Create an atmospheric experience that reinforces the story

3. **The History of Veniss**

The history of Veniss is a long and winding one. Ten years prior to the time of the book “social planners” ruled the city. I believe that the social planners were a form of authoritarian artificial intelligence. Quin helped the government during this time, running an artificial birth program, vat-growing human beings². During this time the city was known as Dayton Central. The government collapsed after a violent insurrection.

The artificial intelligence(s) were defeated and the city was renamed Veniss. The “social revolution” began, bringing about a new appreciation of new, “living art”, free trade, and the death of all forms of “old art”. With free trade came the privatization of the police, large amounts of waste, and pervasive advertising.

Veniss, like all cities in the book, is a self-contained city state, walled off from the rest of the world. Human wastefulness has destroyed the world outside the insulating wall of Veniss (and other city states) and all that remains is a chemical desert. Veniss, which originally had active relations with Balthazar, the nearest other city state, has increasingly become detached from the rest of the world.

A once flourishing off-world mining colony operation has faltered to the point of only barely providing subsistence to the colonies themselves. The Canal District, an area of Veniss full of ocean canals that spaceships bringing goods from off-world used for “wet-landings” is entirely unused by the colonies now.

The city itself is slowly collapsing, moving always further towards barbarism and chaos. “The Bastion”, where Nicola and rest of the city’s programmers work, is the epicenter of the

only real concerted effort to keep the city running. Programmers at the Bastion work to keep the trains and other automated social services in continued operation. They use dated technology that still employs the artificial intelligence that once dominated the city, but it is “carefully yoked for obedience.”³

Much of this state of affairs is due to the fractured and disorganized nature of the city’s government. The city is ruled by many disassociated pockets of authority. For instance, when Nicola travels to Nick’s apartment we learn that “a trip from one end of the city to the other requires eighteen security stops.”⁴ Furthermore, this governmental disorder, and the revolution that led to it, has caused a complete break between Veniss Above and Veniss Below. Veniss Below, originally created to mine deep into the Earth’s surface in order to bring up raw resources to the surface, has become lawless and disconnected to the surface. The mining operations that continue have no relation to the surface and people literally live out their entire lives in the hell of the underground for no reason whatsoever. A long standing lottery system, which allows a certain number of individuals to come to Veniss Above every year, continues to operate but individuals who do win the lottery have no way of telling their families the truth of the situation once they have learned it.

And throughout all of this, Quin and his “Shanghai Circus” cast a long, black shadow across the city. His control above ground is powerful but limited, mostly supported by his continued patronage of the city through gifts of his “living art” (meerkats and ganeshaas). However, below ground Quin is practically a god, controlling nearly all levels absolutely.

4. Quest Story

The player assumes control of Nicolas while she is working at the Bastion. After a brief, textual exposition the player is allowed control of Nicola. The player can interact with Nicolas’ computer terminal (through dialogue) which will allow them to find out the necessary information about Nicholas and his current condition. After finding out that he hasn’t been

financially active for 5 months the player can hack into his bank account, learning that he has recently been issued a new bank card and that he has made a purchase with it on the 10th level underground.

The player then visits Nick's apartment, finding various clues which add intrigue and further the story. After finding a poem about Quin in Nick's apartment the player seeks out Shadrach, the only person he/she knows that has a connection to Quin.

The player visits Shadrach, an ex-lover and current employee of Quin's, at the Canal District. He is forthcoming and lies about the last time he has seen Nick. As he seems to be ready to tell the player something useful about Nick a meerkat appears and they leave together, hinting that the meerkat was actually shutting Shadrach up before he said something incriminating.

The player goes to their (Nicola's) apartment where they are greeted by a ganeshaa and a meerkat. The ganeshaa is delivering the meerkat and the player accepts the meerkat, a gift from Quin, against their better judgment. The player discusses Quin with the meerkat, Salvador, and he recites lines from the poem found at Nick's apartment, information that he should have no way of knowing.

After more suspicious behavior the player follows Salvador to a secret area in the city where Quin's animal creations congregate. The player learns that Salvador and the other creations hate human beings for their cruelty and he/she sneak home before being spotted. A confrontation occurs when Salvador comes home and knows where the player has been by the smell. Salvador tries to kill the player but is nearly killed and incapacitated. Directly after this fight Nick shows up and strangles the player.

5. Level Design

1. General Level Overview

The level begins in the Bastion and progresses from many different but small areas until finally concluding in Nicolas' apartment. Nearly all of the level should have a gritty, urban decay feel contrasted by annoyingly ever-present advertising. The only area to truly break from this feel will be the Veniss_Forest area. The area should predominately use a custom made texture set dominated by concretes, trash, and harsh advertising.

2. Sub-Levels

Nicola's Cubicle

[Veniss_Cubicle]

This sublevel begins the game for the player and will, with the help of expository dialogue, introduce the player to the game setting. Currently, the level will consist of just the workspace of Nicola's cubicle within the Bastion and nothing else. The main object to interact with will be Nicola's computer terminal where she does all of her daily work and the player will learn of Nick's current financial situation (the major initial source of intrigue).

Nick's Apartment

[Veniss_NickApt]

After discovering intriguing information about Nick the player will leave Nicola's cubicle and "travel" to Nick's apartment. This sublevel will consist of Nick's apartment in its entirety (only a bathroom, kitchen, and living room) and will provide major clues to Nick's current status.

[Possibly include walk to and from Nick's apartment (at least the building) so that we can have the player get ambushed on their way back to the train station area. This would allow

us to introduce the fighting element of the game within the first 15 minutes in an acceptable manner.]

Canal District

[Veniss_Canal]

After discovering the poem clue that establishes a connection between Quin and Nick the player will “travel” to the Canal District, in search of Shadrach. Currently, the level will only consist of small portion of the district, overlooking one of the polluted canals where Shadrach often spends his time.

Nicola’s Apartment

[Veniss_NicoApt]

After the player receives little significant information from Shadrach (other than that he is being defensive and vague about the entire subject) the player will once again “travel” to Nicola’s apartment. This sublevel will consist of the entirety of Nicola’s apartment (which is significantly nicer than Nick’s) and will be the site of the climax of the first act. Key features involve a realistically futuristic style and a believably “lived-in” feel.

Alleyway\Secret Forest Area

[Veniss_Alleyway_Forest]

This sublevel will be divided into two areas; the very small alleyway and the much larger forest area. The only key feature of the alleyway is that it’s back wall is false (a hologram) and that the player can pass through it by simply walking to the other side of it. The forest area, on the other hand, needs to be significantly more detailed, as a major plot point and thematic feeling are established within this area. However, we feel that much of this area can be created through the use of existing outdoor tile sets and will hopefully require little effort to perfect. Key features of

the area consist of a bridge over a mudflat, some sort of holographic (or two dimensional) movie screen, and a strong feeling of a pristine, fantastical area somehow sequestered from the urban decay surrounding it on nearly all sides.

3. **NPC List**

- Nicola
- Random Veniss denizens
- Shadrach
- Salvador
- A generic Ganesha
- Variations of Meerkats and Ganeshas
 - Including young and old models
- Nick (normal)

6. **Needed Art Assets**

(See: entire NPC list above)

Geometry tileset for each area

Misc. placeables

7. Design Concerns

What will make the Veniss Above area stand out and really come alive is an implicit incorporation of the major subjects that Vandermeer explicitly talks about. We need meerkats mixing with crowds of humans, a crumbling city almost ironically covered in gaudy and obnoxious advertisements, signs that the last 20 years have been physically traumatic to the city (crumbling structures, damage gone un-repaired, spray painted walls concerning different conflicts, etc) and an overall sense that the city is entirely cut off from the rest of the world. Additionally, interior areas needs to feel lived in, a little messy, and comfortable enough for someone to want to live in. Nicola's apartment has to greatly contrast with Nick's; Nick's should feel as if something very bad has happened not too long ago; Nicola's should feel warm, but also lacking something truly completely "homey".

1 Placeholder name

2 Both Nicholas and Nicola are a product of this program.

3 p. 36

4 p. 37

Section 3: "Veniss Below" Design Document

Veniss Underground

Veniss Below

9.19.07

1. Latest Changes

- [9.19.07] "Veniss Below" design document created
- [9.23.07] "Quest Story" section made more readable
- [9.23.07] "Veniss Below" design document renamed "Veniss Below" (rather than "Veniss Blow 9.19.07")
- [9.23.07] Section 5, sublevels, mostly completed.

2. Goals

- Create a dark, dangerous, and gritty feeling underground that's full of desperation. Also should feel like a machine tirelessly toiling away, all the while oblivious to the fact that its master is gone
- The level begins with Veniss_Ellington and concludes with either Veniss_Entrance or Underground_Rafter
- Reintroduce Shadrach, advance the story through the second major quest, "Save Nicola".
- Get the player further immersed in the story
- Create a feeling of cohesion between the Act I and Act II while exploring the contrasts between the two parts of Veniss

3. Quest Story

The player assumes control of Shadrach while he is visiting “Lady” Ellington, a rich client of Quin’s. During their inane conversation Shadrach realizes that Ellington has a few recently transplanted body parts, including a new hand, eye, and ear. Shadrach realizes that these parts are from Nicola after recognizing a distinctive birthmark of Nicola’s on Ellington’s new hand.

After this realization he goes to Quin’s, almost positive that Quin sent him to Ellington’s estate with the express purpose of seeing what had happened to Nicola. He goes into Quin’s back office and finds out Nicola’s donor name and what bank Quin has sold her to. He learns that she is on the fifth level, in the Slade Organ Bank.

Before heading to save her he inspects her apartment. He finds that it has been ransacked and that Salvador, who we thought Nicola had killed, is still alive. Because of his knowledge from working with Quin he immediately recognizes that Shadrach is an assassin model meerkat, confirming that he was always intended to kill Nicola. With the knowledge that Shadrach is an assassin model, and that that particular model can live for days without their head, Shadrach incinerates all of Salvador’s body but his head. He glues Salvador’s head to a plate and renames him John the Baptist.

He takes John underground with him, entering through a manhole in an alleyway. This allows him to bypass the first two levels of the underground but also leads to his first major fight. Shadrach is attacked by four “little people” who, when defeated, jump off the cliff they are fighting, parachuting down into the abyss. Shadrach then triggers an elevator they were fighting near which falls down after them, crushing them to death.

Shadrach arrives at the fifth level and is confronted by a four mile long line leading to the Slade Organ Bank. Shadrach pushes through this line of “donors” and buyers. He is only allowed to do so because he has a badge from Quin and a gun, but even this is not enough by the time he reaches the head of the line. He has to physically force himself through protesting people until he arrives inside the organ bank.

He is ushered in and witnesses a grotesque sight of two old men fighting over the dead body of a little girl. Dr. Ferguson appears, explaining that the two men are “benefactors” to the bank and therefore are allowed to play with the bodies and do whatever else they may feel like. Shadrach asks about Nicola and Dr. Ferguson looks her up on a computer terminal, learning that she is being held in “live storage”.

Dr. Ferguson gives Shadrach directions to live storage and her body and Shadrach makes his way there. He finds her at the bottom of a mountain of body parts, limbs, etc. He rescues her from the organ bank and takes her to Rafter, a psychewitch, who can hopefully heal her.

After talking with Rafter and confirming that she can help Nicola he leaves to kill Nick. Rafter criticizes him but tells Gollux to lead Shadrach to the 10th level where Nick lives. Gollux guides Shadrach to the 10th level where he must navigate through a winding maze of garbage and battle garbage people who live on this level. Shadrach then finds Nick, barely alive, transformed into a horrible creature by Quin. The player then can choose to either kill Nick, leave him to die, or save him. These three options form the basis of the three alternative endings.

4. Level Design

1. General Level Overview

2. Sub-Levels

"Lady" Ellington's Estate

[Veniss_Ellington]

This sublevel begins the second act of the game and will introduce, with the help of expository text, Shadrach as a character to the player. Lady Ellington's estate is probably very large but the level will only consist of one room, which is opulent to the point of gaudiness. The room will center upon the brief interaction between Shadrach and Ellington and will serve no other purpose than facilitating this conversation and then getting the player out of it as soon as possible.

Quin's Laboratory

[Veniss_Lab]

After making the realization that something terrible has happened to Nicola the player will be transported to Quin's laboratory, in search of information about Nicola's whereabouts. The area will consist of two rooms: one large area with a Quin puppet and caged animals, and one smaller room consisting only of a desk with a computer monitor casting an eery red glow. The player will interact with the computer monitor, learning of Nicola's location (the fifth level underground) and gaining the item "Donor ID#" (a scrap of paper with Nicola's donor ID that will eventually be used to locate her in [Veniss_OrganBank]) The player will also be able to pick up one of Quin's ID badges, allowing the player to go through all areas that Quin controls (item "QuinID").

Nicola's Apartment

[Veniss_NicoApt2]

After learning where Nicola is the player "travels" to Nicola's apartment to investigate what happened to her. This level is a ransacked version of Nicola's Apartment [Veniss_NicoApt] with signs of a struggle. The player learns, from the environment, some of what occurred in Nicola's apartment and, most importantly, discovers Salvador\John the Baptist. The player will acquire John as an item (possibly binding) that can be worn on their wrist.

Alleyway

[Veniss_Alleyway]

The player "travels" to a deserted, trash-filled alleyway where they descend into the underground. This allows them to skip going through the security checkpoints and brings them directly to the second underground level. [Possibly add a fight to make this more interesting, or possibly some sort of puzzle.]

Elevator Platform

[Veniss_Platform]

This area should be darkly lit, gritty and small, only exposing the platform and a nearby elevator (it is inside of an elevator shaft.). The player is transported to this area after entering through the alleyway and after a small amount of exploration is attacked by four little creatures\people. After doing enough damage to them they flee, jumping off the edge of the platform and parachuting into the abyss. After this is witnessed (probably through a textual description of some sort) the player will have the ability to disengage the emergency break on the nearby elevator, sending it plummeting down the shaft, killing the four attackers.

The Fifth Level

[Veniss_5thLevel]

This area should seem claustrophobic, deeply subterranean, and populated by a vastly desperate group of people. The player must follow a long line of "donors" and buyers waiting for their appointment with the Slade Organ Bank. The player should encounter many snippets of stories through the "shout" dialogue of the various NPCs that he/she passes. The dialogue should become more hostile, about the player cutting in line, as the player gets closer to the front of the

line. The player should be stopped, by way of a forced dialogue, and must either use Quin's badge as an excuse (which will be acceptable for the first stop) or fight and kill the person who stops him/her. After this the player will be stopped once more when they get to the front of the line, with the only options being intimidation or violence. If the player does not navigate the dialogue tree correctly it will result in a fight to the death between the player and then NPC who stopped him/her.

Slade Organ Bank

[Veniss_OrganBank]

This area should resemble a cavernous and grotesque cathedral. The walls are covered with inset glass cocoons housing human beings. Massive pillars made up of tubes transporting vital fluids add to the cathedral feeling of the place. The Bank should feel chaotic, dirty, and grotesque. It should also seem paradoxically efficient and wasteful; the operations of extracting organs and essentially butchering human beings is an efficient process but there is waste everywhere (piles of rotting limbs, unused organs, general disarray, etc.). After a brief conversation with a receptionist the player will be allowed to enter (because of Quin's badge) and will meet Dr. Ferguson. Dr. Ferguson will explain the situation and tell the player where they can find Nicola. The player will follow Dr. Ferguson a certain distance and then he will tell them where to go from there. The player will find a mountain of limbs (with Nicola in the center) [we need to find a descent play mechanic for how this whole thing is going to happen.] The player then leaves through the way they came, with Nicola. Through dialogue with Nicola the player "remembers" that they know a psychewitch who can heal Nicola and when they exit the organ bank they "travel" to Rafter's house.

Rafter's House

[Veniss_Rafters]

Rafter, the psychewitch, lives in a meager, cramped "apartment" in the area where the human miners use to live. Her house is carved out of stone with rusty metal accents. The player is transported to her door step and after dialog is allowed to enter. Rafter will heal Nicola and the player then leaves to kill Nick. Rafter criticizes the player but sends her friend, the Gollux, to go with him as a guide.

The Waste District

[Veniss_WasteDistrict]

The player "travels" to this level and must navigate through giant heaps of garbage to find Nick. Along the way they witness the giant metal worm that "eats" the garbage and are attacked by the garbage people. After a while of fighting and navigating through the trash maze the player finds Nick, which begins the conclusion of the game.

3. **NPC List**

- Lady Ellington
- Quin's Puppet
- Four little people\creatures
- Random (roughly 10-20) NPCs in line
- Receptionist
- Dr. Ferguson
- The "Patrons"
- Nicola
- Rafter
- The Gollux
- Nick (deformed)

5. **Needed Art Assets**

(See NPC list above)

Geometry tileset for each area

Misc. placeables

6. **Design Concerns**

Appendix 2: Workflow Tutorial

Maya to NWN

Programs Used:

Microsoft Windows XP
Photoshop (CS3)
Maya (8.5)
3DS Max (5.0)
Gmax (1.2)
PolyTrans (4.1.2)
Neverwinter Nights
Neverwinter Nights Toolset
SetEditor (B085)
Bif Packer/unpacker (1.0)
Notepad (I prefer Textpad)

Scripts/Plugin Packs:

3DS:	NWmax (0.8) MDL Plugin Suite (1.1b)
Gmax:	NWmax for Gmax
Maya:	OBJExport plugin
Photoshop:	Targa plugin (available on adobe's site)

This workflow was discovered after many, many hours of testing. I won't claim that I had no help at all, but there seemed to be no resources with this information online. For the work in 3DS, there are many tutorials that can be found online, even though they are quite old, that I used once everything was successfully imported into 3DS.

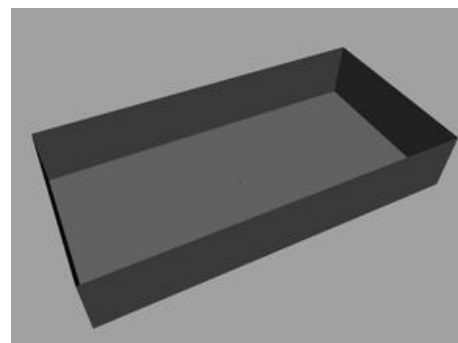
Start off by installing Maya, 3ds max 5.0, PolyTrans, and Gmax (free from TurboSquid⁴). Once those are installed, we can move on. (If you are on Vista, just stop now and find an XP machine, because 3DS Max 5.0 will not work)

Next we need to install the necessary plugins that will help handle NWN file formats. Maya needs the aptly named objexport plugin in order to export .obj files. That can be found online, along with instructions which are as simple as dropping the file into the right folder. NWmax can be downloaded at this website: <http://nwwvault.ign.com/View.php?view=Other.Detail&id=626>. Download both NWmax⁵ versions that you need (Max 5, as the mdlsuite works in 5, not 8), and install according to instructions. Do the same for the mdlsuite⁶. Also found on the vault is the SetEditor, which is a standalone program for modifying tileset files⁷. The Bif packer/unpacker allows you to take the .bif files from nwn\data and unpack them in an alternative directory for access to files used in game (<http://nwwvault.ign.com/View.php?view=Other.Detail&id=607>).

At this point everything is ready to go, so let's begin by unpacking the bif files to a usable format. Launch the BifEditor program and go ahead and hit open key file. We are not looking for any certain object/texture/file, but for the purpose of this tutorial unpack a bif file that includes tiles (it should be indicated in the name). Unpack it to a location of your choice (I keep everything in \Neverwinternights\BIF\). If you are looking for something particular you'll have to unpack the whole lot of them, and rummage around a bit. Go ahead and launch 3DS max and chose File – Import. Select Neverwinter nights model (mdl) from the filetype list and navigate to the folder you unpacked your bif file(s). Import any file, and examine what it looks like. Notice the large hexagon at the base of the model– this is a key part of getting NWN to recognize your tiles. Also notice that there are meshes right on top of each other. This allows for collision detection, and prevents players from passing through tile objects or walls. We will be building from scratch, so you can close down 3DS and launch Maya. From this point forward, you should also open the picture tutorial to help follow along.

BUILD

Now we can start building our own tile. Before doing anything in Maya, a few settings must be changed to keep scaling issues to a minimum. Under Window – Settings/Preferences – Preferences you will see a pane to the left, click on settings, and select centimeter for your units. All other settings can stay the same, including the time setting, as



Objects made in Maya, box with normals flipped, plane

⁴ <http://www.turbosquid.com/gmax>

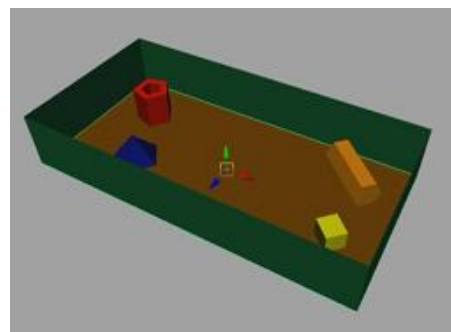
⁵ Note that in order to use the tools correctly, after the NWmax (Gmax) install, you must use the NWmax shortcut, not Gmax.

⁶ <http://www.google.com/search?ie=UTF-8&oe=UTF-8&sourceid=navclient&gfns=1&q=nwmax+download>

⁷ <http://nwwvault.ign.com/View.php?view=Other.Detail&id=620>

I animated in 3DS (this functionality is not detailed in this tutorial). In addition to this setting, in your view pane menu, select View – Camera Attribute Editor. Adjust your clipping plane to something around 100000 (default is 1000). I usually just throw a 9 in front to make it 91000. This will prevent any draw distance issues that may occur when operating with such a large mesh.

Now, each tile in NWN is 10m by 10m, so keep that in mind when you are building an area. Start off by making a ground plane 2000cm x 1000cm (you will want to manually enter this) and then zoom way out to see the whole thing. Make sure the plane is centered at (0,0,0) . Next, we will make the walls for our area. The easiest way to do this is to make a cube 2000cm x 1000cm x 350cm (WxDxH). Select bottom and top faces (F11 is default for face mode) and delete them. Now go back to object mode (F8) and from your normals menu, select reverse. Check to see that the normals are pointing towards the center by going to Display – Polys – Face Normals. To show the effects of a walkmesh, we will now create a few obstacles in our area. Make any shapes you want, but try to keep them within the bounds of the area. Texture the walls, ground, objects any way you like, but keep in mind it may be difficult to replicate what you see in Maya in 3DS (tileable textures are highly recommended, as the UVs often don't translate correctly. Specifically, do **not** use Automatic Mapping in May). Also, NWN does not support multiple textures per object, so ahead of time you may select and extract faces that will need it (walls are a good example of this). For this tutorial I simply applied .tga (targa) textures of solid colors— green, red, blue, brown, yellow, and orange. Do not use procedural textures here, they show up very dull and off-color in NWN.

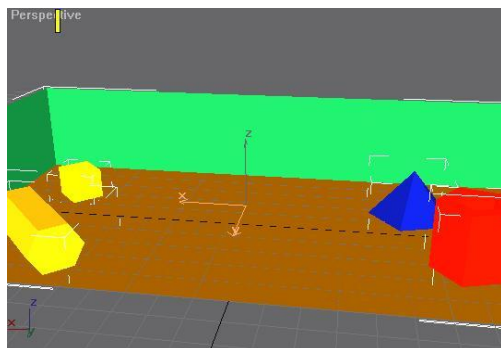


Then textured with random objects

POLYTRANS/GMAX

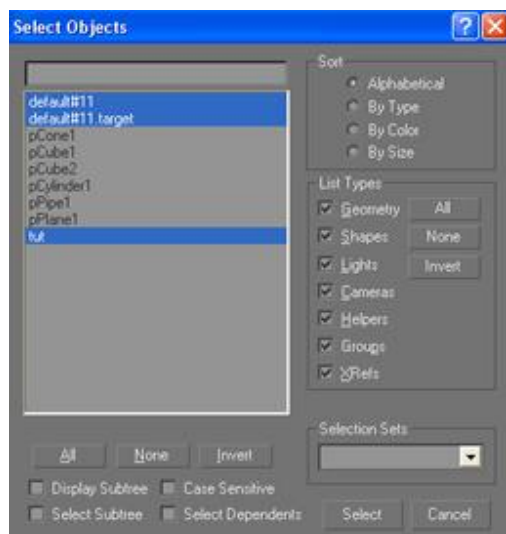
Now in order for Gmax and 3DS to read your files correctly, you need to translate to .3ds file format. Launch PolyTrans, and choose file - import – wavefront object file (default settings are fine). You can choose whether or not to add a perspective camera– we will just delete it in Gmax. Now chose file – export – 3D Studio* (again, default settings are ok).

*If your models are too complicated, there may be missing faces; this is a limitation of PolyTrans



First imported, into GMax, perspective view

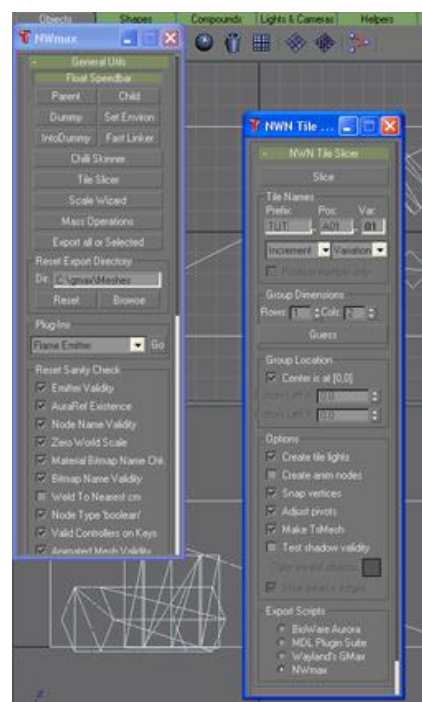
Now launch NWmax (will open Gmax), and chose file – import. Locate your new .3ds file fresh out of PolyTrans, and import. Texturing may be stretched or inaccurate, or not be there at all, but it's okay, we are going to reapply the textures later through 3DS.



Select by name, delete

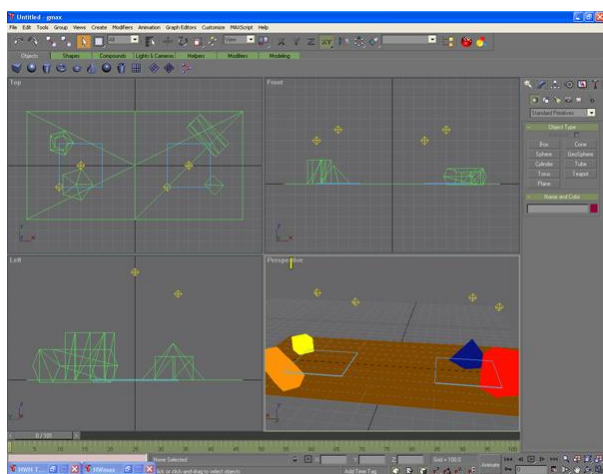
The first thing you will do is click on the select by name button (mouse pointing at a list). Select the two default items, as well as the item that is named the same as your 3ds file (mine was tut, from tut.3ds). Delete these three items, as they are useless.

Next, you can use the NWmax panel to slice your area into tiles (in this case 1x2). On the NWmax panel, choose general utilities – tile slicer. Highlight every object and the go back to the tile slicer panel and hit guess under group dimensions. “1x2” should be generated in the rows/columns area. After that, give your tiles a three letter prefix, and go ahead and hit slice. It should be very quick because our models are simple at this point.

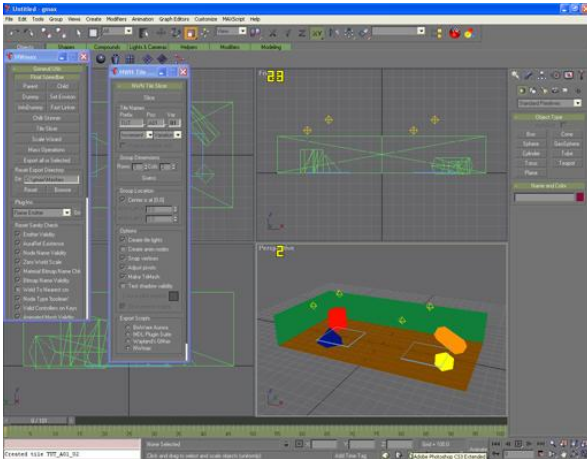


**Highlight all, and open tile slicer.
Follow instructions, and hit slice**

You should see some extra objects in there now, but oh no, the walls are gone. They turned out to be too close to the tile edge. Use Ctrl+Z to go back to the pre-sliced version.



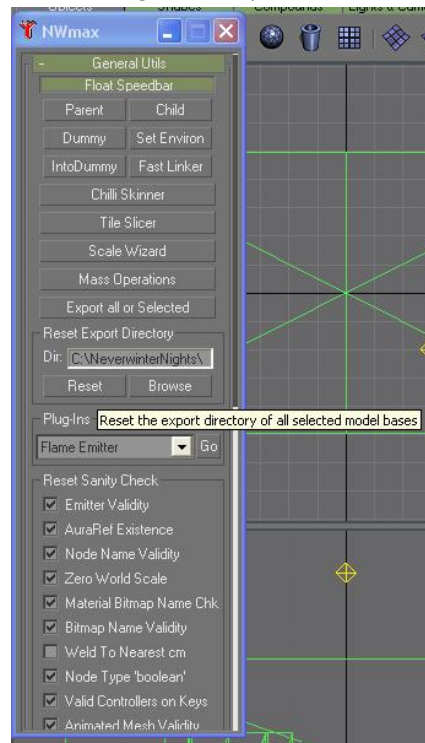
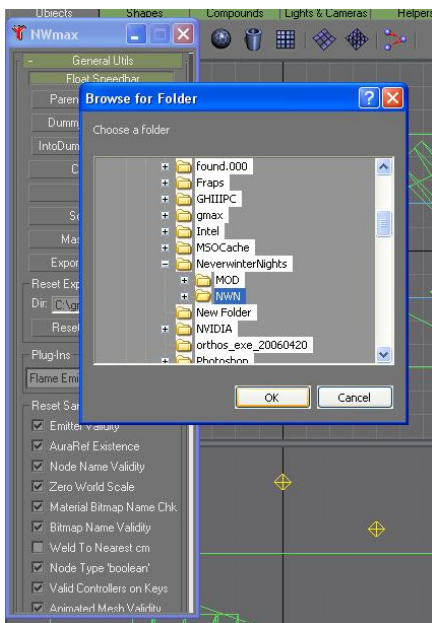
Sans walls



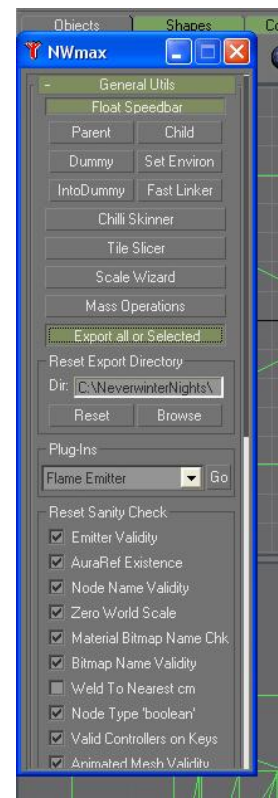
Slice again, success

Again, select all objects, and then chose the uniform scale tool, which looks like a box inside a box. Click on your object, and drag till all three x, y, and z coordinates at the bottom read 99. That should shrink things just enough to get the walls in there. Now go back and slice again. Sweet, sliced and it has walls, now we are almost done with Gmax.

Go back to the main NWmax panel and under General Utilities – Reset Export Directory, hit browse. Navigate to your override folder for your NWN installation, something like c:\neverwinternights\nwn\override (don't use the keyboard; use the mouse to navigate the menu or it does not work right). Now highlight everything, hit the reset button next to the browse button you just used, and then hit the “export all or selected” button right above that box. The tiles should export quickly, and to make sure go check your override folder. If all looks right, then it is now ok to exit Gmax, and probably save a copy just in case.



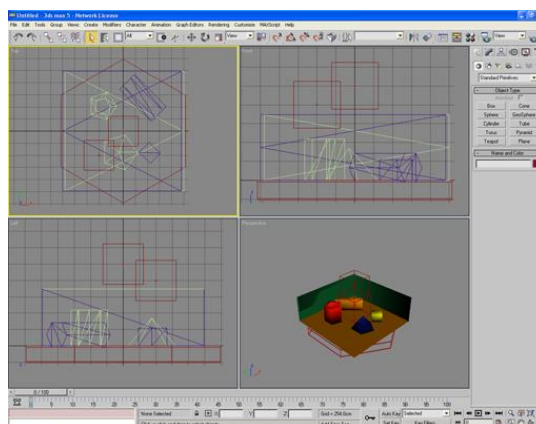
Reset



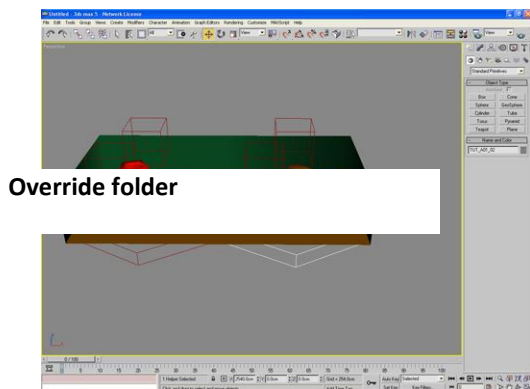
Export

3DS

At this point we need to open the tiles in 3DS for further work. Launch 3DS and select file – import. Navigate to your override folder, where you exported your tiles from Gmax. Select Neverwinternights MDL from the dropdown to see your tiles. Import your first tile, and then repeat to import your second tile right on top of your first.



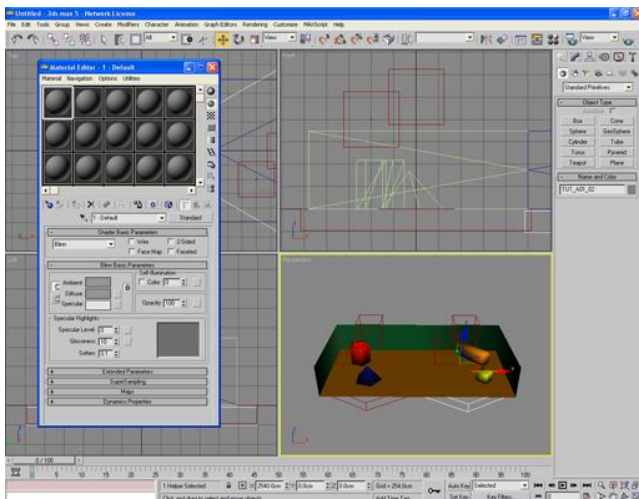
Imported, stacked



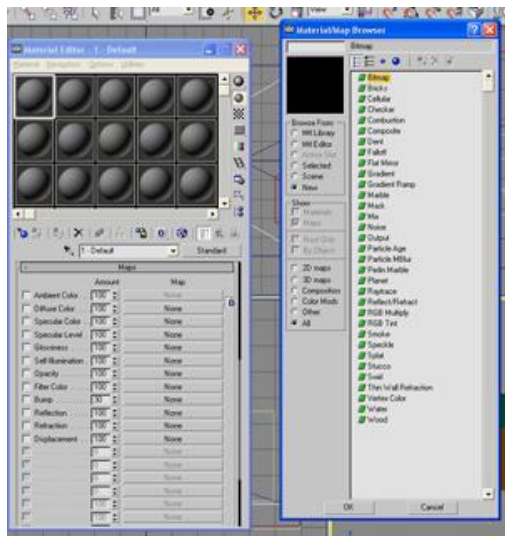
Override folder

Use the select by name tool to select the model base for your second tile. Slide your tile base 2540 units in the positive x direction (sometimes it is 1000, not sure when or why though).

Slide one tile over



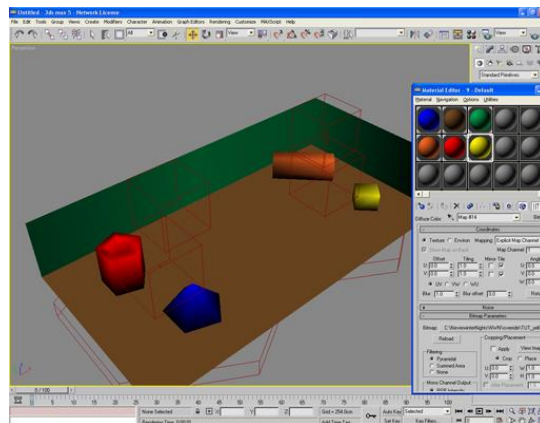
You should see your entire area together, textured and all. The problem is your texture references (UV map) are gone (to NWN at least), so you will have to reapply them. Before doing that, we need to put the textures in a place NWN can find them, namely, the override directory. Locate your texture files and copy them into the override directory. Now bring up the material editor in 3DS (hotkey m), and highlight the first grey sphere.



This is just like hypershade (the Maya material creation program), and is not too difficult to use. Down below there is a maps button, hit it, and then select the “none” box next to the diffuse color option. Navigate to your override folder, and select the first of your textures (I suggest at this point you rename your textures with prefix of your tile, for example my tiles are named TUT_A01_xx, and the textures are TUT_texturename).

We can now apply the texture. Select the bottom right-most button in 3DS to expand the active view to full panel (this helps with more detailed work). In your materials pane, with your newly imported texture selected, click the checkered cube underneath the material display. This will allow you to see your texture on your objects without rendering. In your regular view, select the object you want to apply the texture to, and go back to the materials pane. Hit the apply material button in the materials pane (arrow pointing to a box). The shading of the object may change subtly, may not, but to check if it is applied, hit F9 for a quick render, and see that the material is not listed in the missing textures. Repeat these steps for each texture.

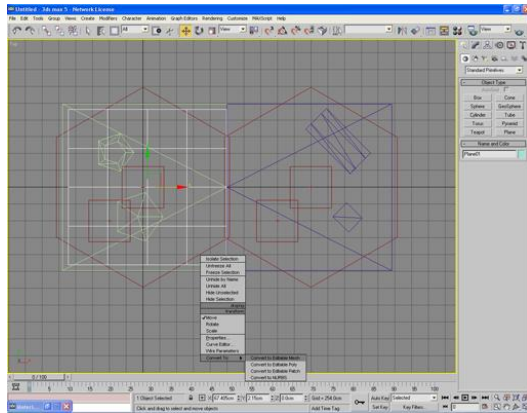
* (Due to NWN limitations, objects can only have one texture applied to them. If they were auto mapped in Maya, they may not align in 3DS)



Textures import and applied

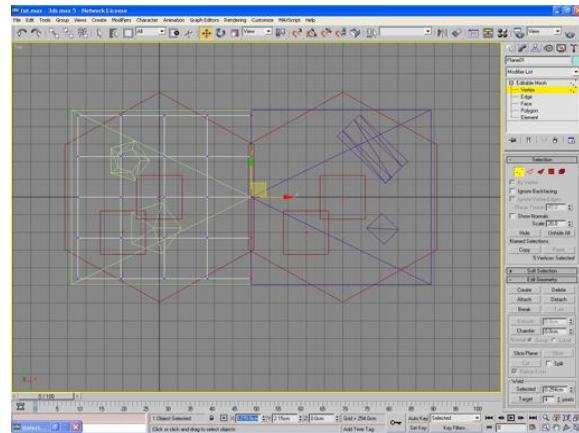
Now these tiles would get into NWN just fine, but you will not be able to place a starting point or even walk on its surface. In order to navigate on the tile you need what is called a walkmesh. This determines where you are allowed to travel on a tile. Having the MDL suite installed correctly is a necessity here, as you need to have it to create the mesh. Start by changing your view to the Top view, and keep the render mode in wireframe. To the right is a large pane, by default the first tab is the create tab, which we want. Under the create tab, select plane, and make a plane slightly smaller than the outside walls of ONE TILE (not the entire area). Now looking at my area, there will be some narrow paths, so I may adjust the number of quads in my plane in the right pane, under length and width segs. Due to NWN’s dated engine, be wary of the number of polys you are using here (we don’t want to be wasting thousands of polys on an invisible object). Right click anywhere in your panel (the plane should still be

selected, and from the menu choose convert to... - editable mesh. This will change your right pane tab to the modifier tab. In this tab, expand the editable mesh tree and select vertex.



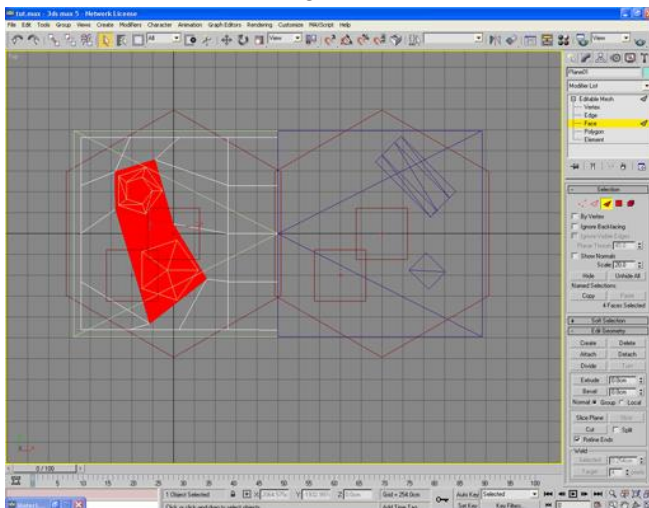
Convert to Editable Mesh

Highlight the vertices along the right side of the plane, and select the move tool (top toolbar). Move these vertices to position 1270 on the x axis. Now grab the top right vertex and make the y position something easy to remember (mine ended up being 1175). Do the same for the bottom right (-1175).

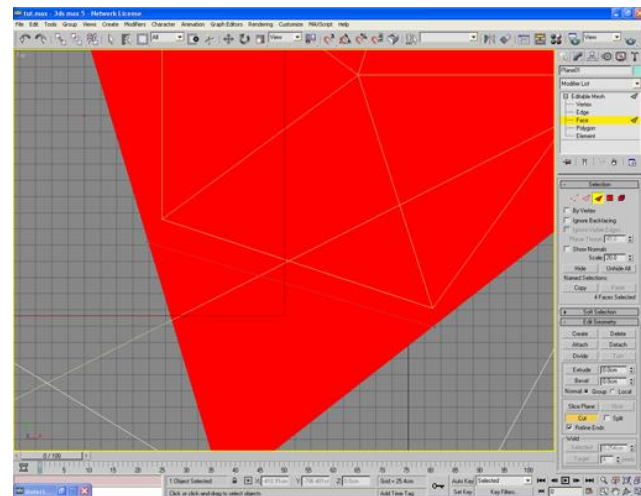


Slide vertices

Now to make holes in the mesh to allow our objects to be maneuvered around, we will shuffle around vertices. Using the move tool, move the vertices around (NOT vertical for these purposes). As you can see, I've run into a problem already. There would be a large gap in between my objects. To get around this, select face in the modifier tab, and select those faces that need cutting. Select the cut tool from the lower half of the right pane, and slice the faces to fit your needs (cutting is done by getting very close to one edge, clicking, and doing the same for the end of the cut). If there are any awkward shadows, that means there are overlapping faces, and you must move vertices around to solve the problem. This can be fixed after slicing faces.

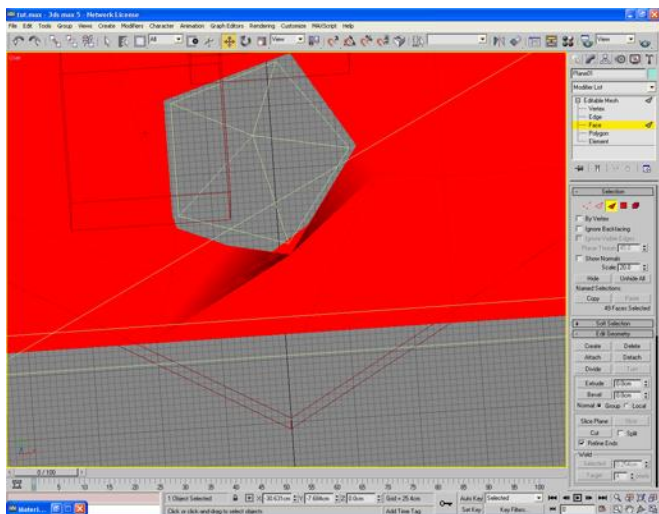
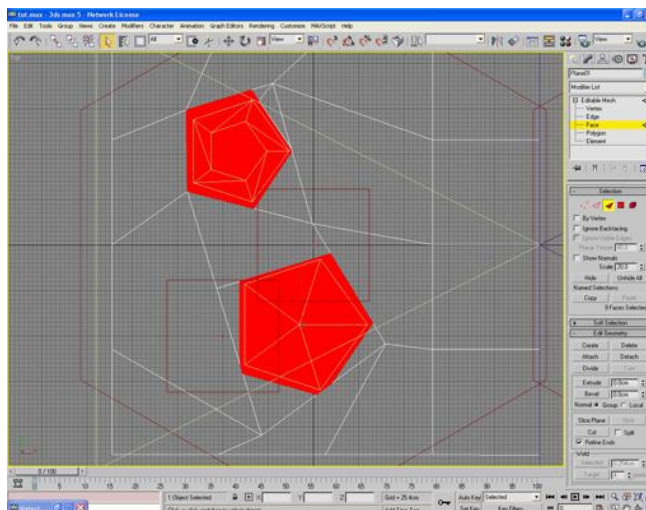


Selected faces to be cut



Cut faces to accommodate objects

Once sliced, select all faces that are under objects, or mark the bounds of your object, and select delete from the right pane.

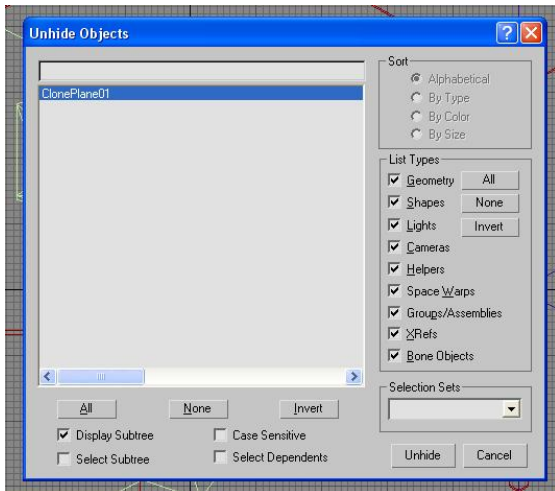


With face selection still active, highlight all faces, folds will now become apparent. A bit of messing around and the folds are gone, and we may continue.

On the far right pane, select the tools tab, it is the last tab there by default. Hit the MaxScript button, then run script, then navigate to \scripts\NWmax\CS3.02\ and run the script in that folder. This is the chiliskinner, and we will only use a single button here.



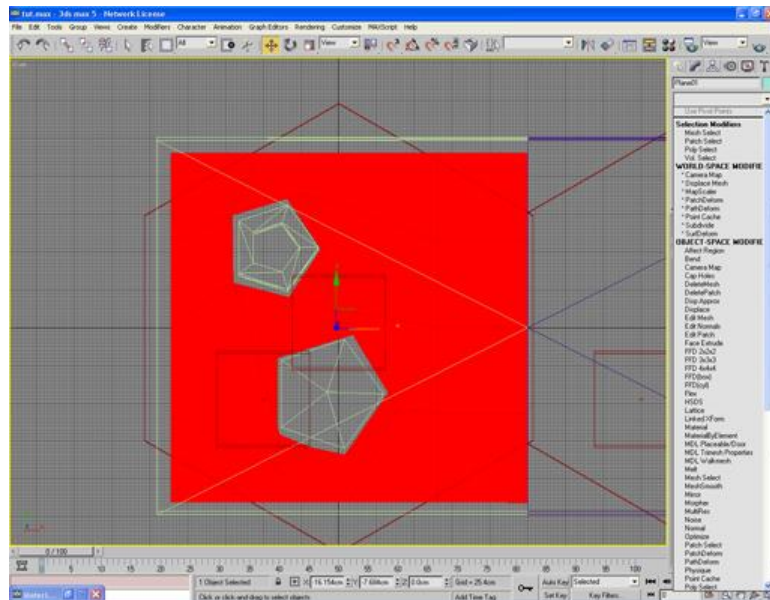
Chiliskinner



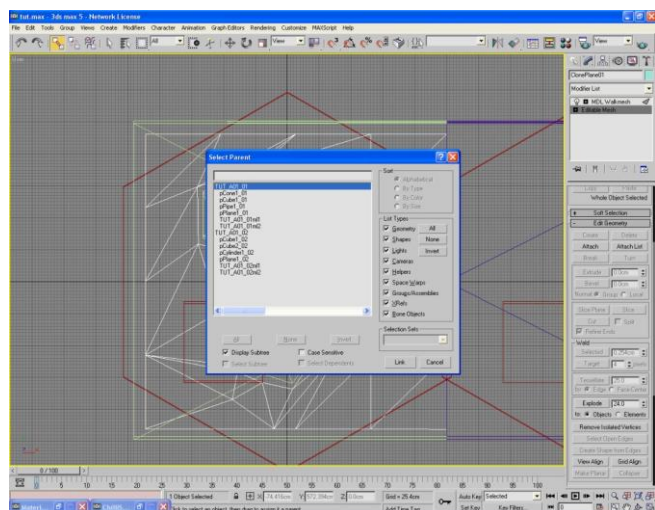
Clone, and unhide by name

With your plane still selected, find the clone box on the chilskinner interface, it looks like a colorful cube with a 2 on it. Hit it, and everything will be selected after the operation. Click somewhere to lose focus, and then select your mesh again. Delete the whole object (Yes, seriously). Now right click in your pane and select unhide by name. Unhide your cloned mesh.

To turn this into a walkmesh, we must add a modifier. There is a long dropdown menu of modifiers at the top of the modifier pane, from this list, select MDL Walkmesh. Now expand the walkmesh tree and select face or polygon, and select every face. On the modifier pane there are material types, select the material best suited for the area. Once highlighted, click on the editable mesh tree, and ignore the warning that comes up, and hit YES. Last thing before you can continue to the other tile, you must link this mesh with the tile base.



Check which tile you are working on by clicking the large base (hexagon) and looking at the name in the modifier pane. Once that is determined, select your mesh and click on editable mesh in the modifier pane (hit YES again). Hit the link button (two boxes with a chain), and then the select by name

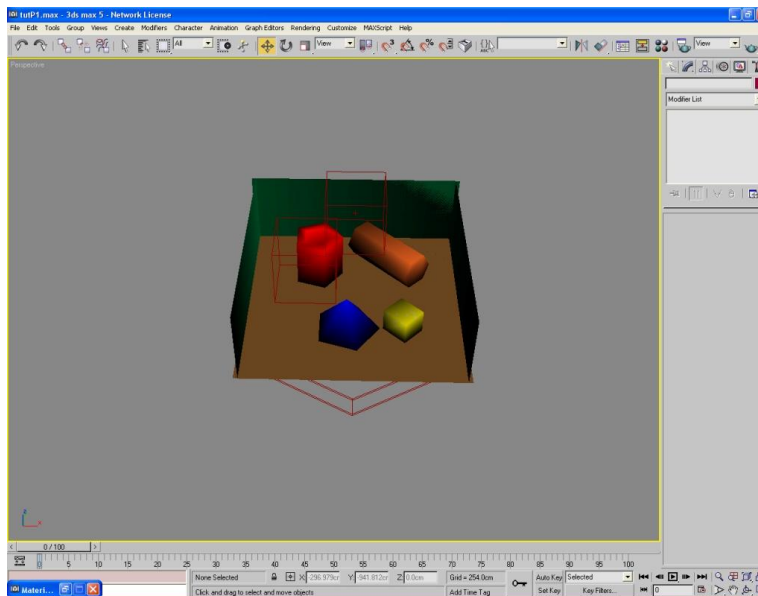


Parent mesh with base

tool. Select the correct model base and hit link. Immediately from here, before you go linking stuff unknowingly, choose the select tool (regular mouse icon). To make sure everything is linked, open the select by name tool, and in the options select display subtree (a good idea anyhow). Make sure your cloned plane has the right parent.

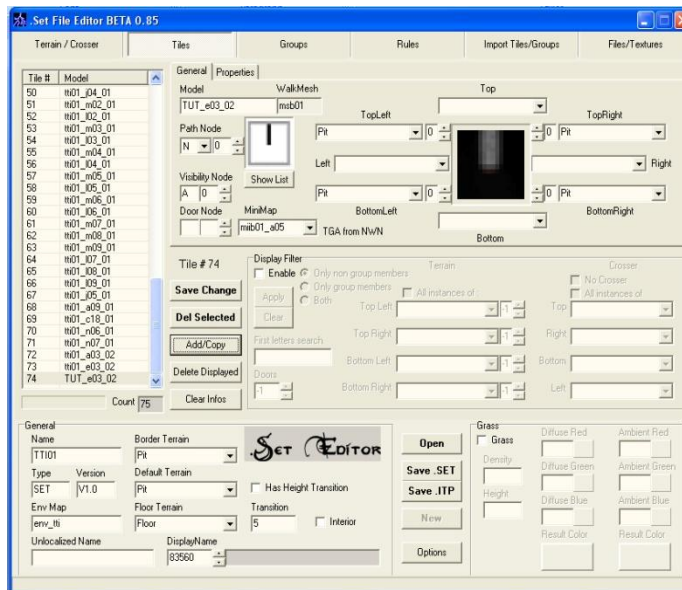
Do this whole thing again to the other side, making sure that the vertices are at the same location in between tiles. In this case they should be at 1270. Also remember to move those corner vertices to the correct y position 1175/-1175.

We now want to go to select by name, and select only the two model bases. Using the move tool, move both to location (0,0,0). It will look a bit silly, especially when you are working with many more tiles. Now we can finally export everything. Open the select by name tool, now check off the box that says select subtree. Click on the first model base, and everything linked to it should be selected as well (make sure nothing is floating around by itself as well). From the file menu, select export selected, and save the tile with the appropriate name (TUT_A01_01.mdl). This will generate another file as well, the walkmesh file (.wok). The defaults on the exporter are fine. Overwrite the earlier version. Select the subtree from the second modelbase, and export as the next tile (TUT_A01_02.mdl). Check in your override folder that the time on the file has been updated, and go ahead and close 3DS.



Move bases to (0,0,0), export

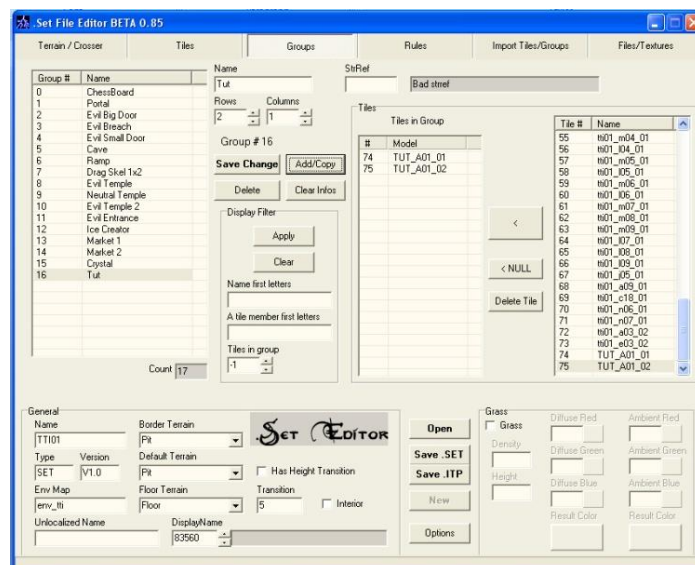
ALMOST THERE



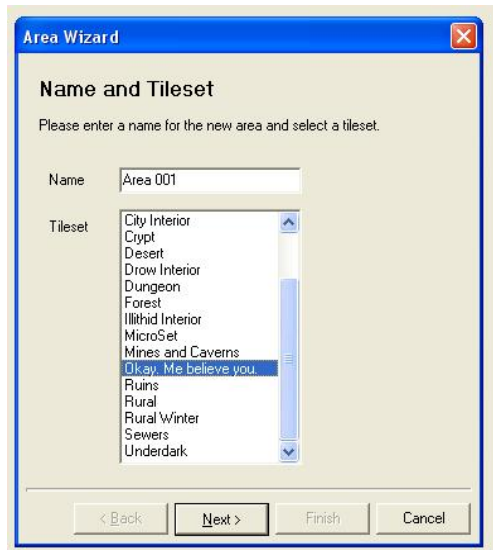
Adding your tiles

Launch the .set editor you downloaded earlier. Click open in the editor, and navigate to where you unpacked the bif earlier. Open one of the .set files. I will be using the tt101.set (from xp2_tiles), but they all have similar structure. If you are building interiors, the tileset name usually has an I in it. Now save it into your override directory before doing anything. First save the .set file, then save the .itp file. In the second tab, select the very last tile. Change the name to *yourprefix_A01_01*, set the path node to N, change the topleft, topright, bottomleft, and bottomright areas to Pit (or the default empty space for that tileset), and the minimap to anything really, I chose what looked like N (miib01_a05). Hit the Add/Copy button to add your tile. With your tile selected, change the last number to 2 for the second tile and hit add/copy again.

Move on to the groups tab, and type in a group name. Select, 1 x 2, and add your tiles, in order, to the group tiles. Before saving and exiting, change the display name of the whole tileset (bottom left area) to anything by using the up and down buttons (I hit down twice to get some random quote). This will be the name of the tileset in the toolset. Now save both the .set and .itp again.

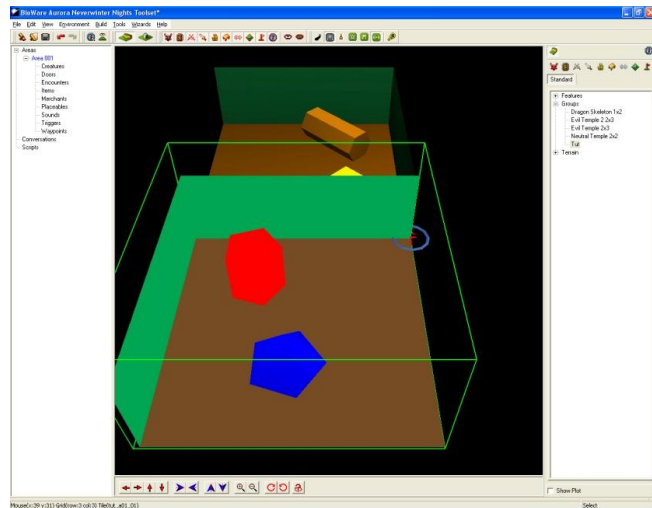


Create a group of your two tiles



Okay, Me Believe You

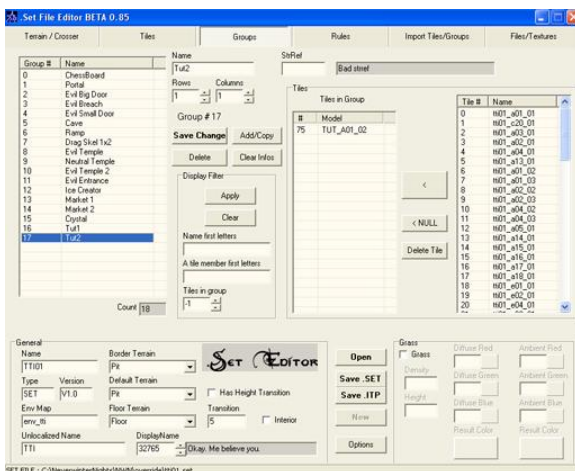
Launch the toolset. Create a new area, chose your oddly named tileset (mine was “Okay, Me Believe You”). Find your group, and place it down.



Bad rotation

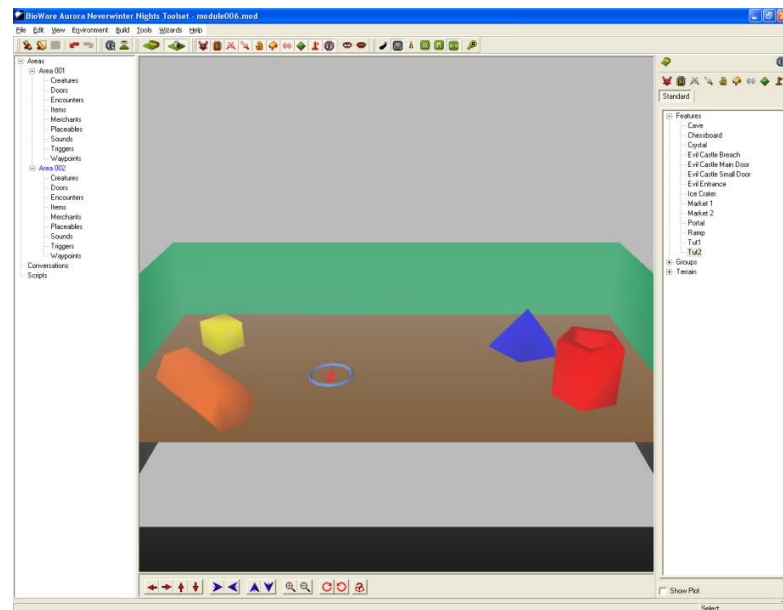
You may have rotation problems, but luckily there is an easy work around.

Simply open up the set editor, and open your tileset. Change your group you made earlier to a 1x1 (remove second tile first), and rename *groupname1*. Create a second group with the other tile named *groupname2*. This allows for separate placement of the two tiles, which you can rotate in the toolset with a right click. Restart the toolset to see these changes. Find these new single tile groups under the features tree, not group.



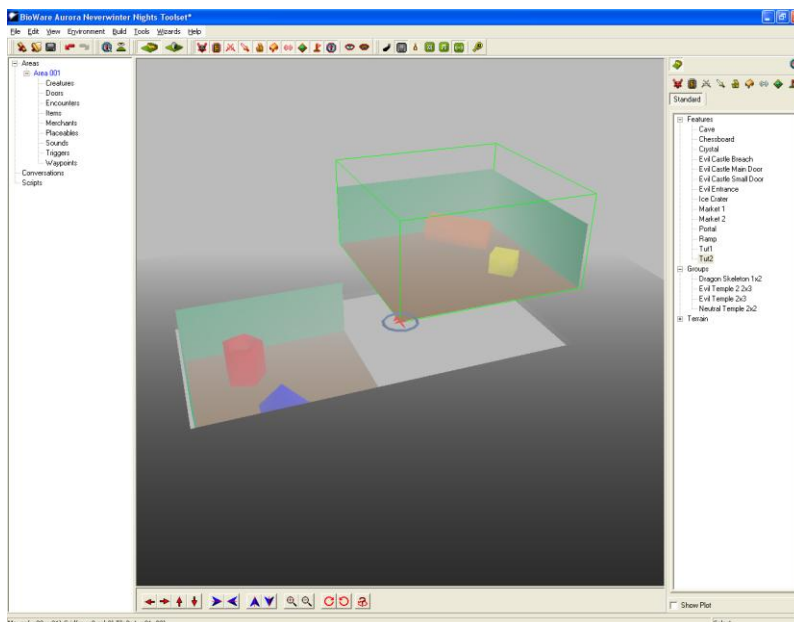
Regrouping

You will notice now that your tiles line up, and although elevated, you can place the start node on your tile. Now from the toolset hit F9 to test everything out. Success! Everything should work great. You can't walk through objects, and you may notice that the walls disappear when you are behind them, that is because the normals are facing inwards.



Looks good

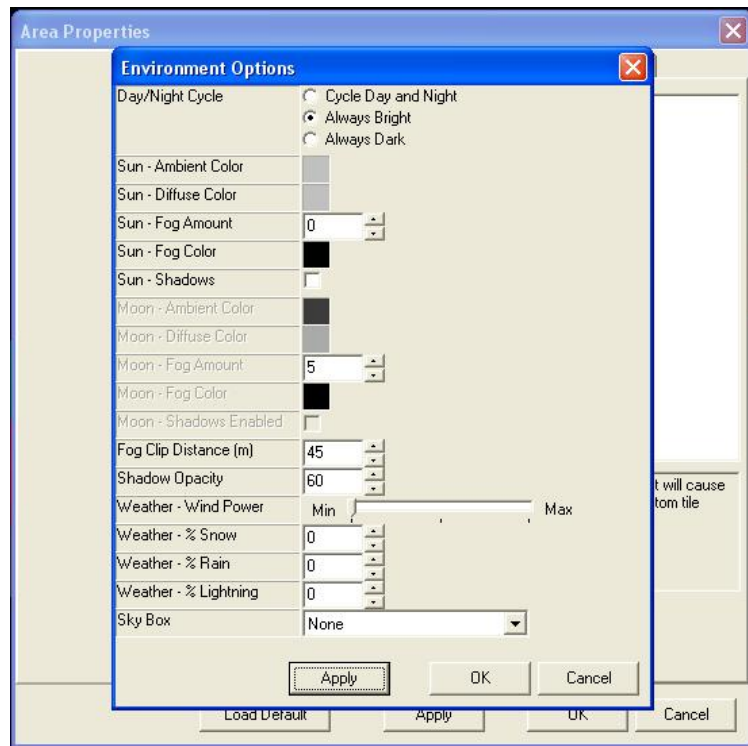
My test showed a small gap in the walkmesh on the tile seam, but that is easily fixed by checking that vertices' height and x position, and if it is still at 0 and 1270, than I slide it 1cm into the other tile to get an overlap with the other walkmesh. That fixed that problem. Now, if you tiles are still floating in the air, let's get rid of that by one of two methods (see walkmesh creation pictures for reference).



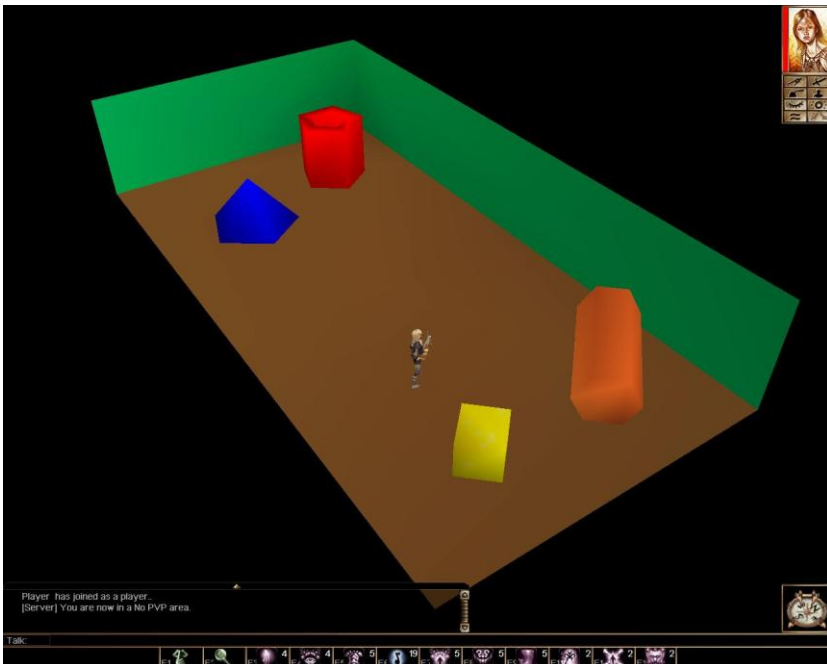
Almost

The hard one is by shifting our tiles. Import your tiles into 3DS, and select everything but the base, group them together (group file menu), then shift down around 900 (I am guessing here). Ungroup the objects and then export again. As you can see that just about did it for one tile, but there is a work around for this as well.

Much easier is the second method. In the toolset, select edit – area properties, and select the visual tab. The default for this tileset was exterior snowing, which does not really fit, so change to interior bright, select always bright, and change the sun fog to black, and the ambient colors to something much lighter. Hitting apply will show you a preview before you close the window so this is pretty easy.



Adjust area lighting to mask gray



Enjoy working area!

Hit F9 again to test your area in NWN, and be mindful that the default character (the first one listed) is not of normal height. Keep testing the area until there are no collisions or random jumps. Finally, everything is done and you have a working area.

The most common problem I had in this discovery process was that I would suddenly be way up in the air when I tried to walk on my tile, or in between. This means that the particular area is missing a walkmesh, or that there is a gap between two tiles walkmesh. I have successfully used this method on things up to 2x4 tiles, and everything works just fine. Remember to parent your objects correctly in 3DS, otherwise they will not display in NWN. Also, if you create any of your objects once in 3DS max, be sure to convert them to editable meshes. Not only does that make it easier to work with your objects, but it is necessary for NWN to read them correctly. Making placeables is much easier, and should be used to fill in most of your tiles, that way you can keep the tile geometry nice and clean, and uncluttered.

Sources:

Mr. X . Tutorial: Creating Custom Tiles, NWvault,
http://nwwvault.ign.com/?dir=resources/tutorials/&content=tile_tutorial

Bioware Corp. "Tileset Construction." Copyright 2002, Infogrames Entertainment. New York, NY.
[Obtained from Bioware's own website as a free download to registered users.]

Christopher. Bioware custom content forum post. Tuesday, 26 July 2005 03:06PM.
<http://nwn.bioware.com/forums/viewtopic.html?topic=162084&forum=48&sp=45>

Appendix 3: Sample Scripting

- ```
//alley_port

//Written by Garon Clements

//Script to teleport PC to the Forest Area when they click

//on the door switch in the alley

void main()

{

 object oClicker = GetClickingObject();

 object oTarget = GetTransitionTarget(OBJECT_SELF);

 object oPC = GetFirstPC();

 SetAreaTransitionBMP(AREA_TRANSITION_RANDOM);

 AssignCommand(oClicker,JumpToObject(oTarget));

 AssignCommand(oPC,ActionJumpToLocation(GetLocation(GetWaypointByTag("WP_ForestToAlley"))));

}
```
- ```
//alley_switch

//Written By Garon Clements

//Script to create hidden door switch when PC walks

//into trigger area
```

```

void main()
{
    CreateObject(OBJECT_TYPE_PLACEABLE, "HiddenDoorSwitch" ,
    GetLocation(GetWaypointByTag("hidden_switch_point")) , FALSE);

}

```

3. //canal_expo
 //Written by Garon Clements
 //Script to initiate expository dialog in the canal area

```

void main()
{
    object oEnter = GetEnteringObject();
    if (GetIsPC(oEnter) == FALSE) return;

    object oNPC = GetObjectByTag("Canal_Expos");
    AssignCommand(oNPC, ActionStartConversation(oEnter));
}

```

4. //canal_scene
 //Written by Garon Clements
 //Cutscene used in the canal district
 //Created using in_g_cutscene engine


```
#include "in_g_cutscene"

void main()
{
    //Find Actors
    object oPC = GetFirstPC();
    object oShad = GetObjectByTag("Shadrach");

    //Find Waypoints
    object oMW1 = GetWaypointByTag("meerkat_point1");
    object oMW2 = GetWaypointByTag("meerkat_point2");
    object oSHDW1 = GetWaypointByTag("shad_point1");
    object oPCW1 = GetWaypointByTag("pc_point1");

    // Start the cutscene
    GestaltStartCutscene (oPC, "canal_scene");

    //Create Meerkat
    GestaltCreate(0.0, oMW1, OBJECT_TYPE_CREATURE, "meerkat", "Cut_Cat", FALSE, 0,
FALSE);

    //Move Meerkat to Shadrach
    GestaltTagActionMove (1.0, "Cut_Cat", oMW2, FALSE, 0.0, 5.0);

    GestaltActionFace (5.0, oShad, 0.0,2,oMW2);
```

```

//Turn Meerkat and Shadrach around and move them away from PC
GestaltTagActionFace (6.0, "Cut_Cat", 0.0,1,oMW2);
GestaltActionFace (6.0, oShad, 0.0,2,oSHDW1);

GestaltTagActionMove (6.5, "Cut_Cat", oMW1, FALSE, 0.0, 5.5);
GestaltActionMove (6.5, oShad, oSHDW1, FALSE, 0.0, 5.5);

//Turn PC around and walk away from others
GestaltActionFace (8.0, oPC, 0.0,1,oPCW1);

GestaltActionMove (8.5, oPC, oPCW1, FALSE, 0.0, 4.5);

//End Cutscene and Teleport PC to Nicholas Apartment
GestaltStopCutscene (13.0, oPC, "NicholasBed");
}

```

```

5. //close_computer
//Written by Garon Clements
//Script used to play computer close animation when PC
//logs off.

void main()
{
    PlayAnimation(ANIMATION_PLACEABLE_CLOSE, 1.0, 0.0);
}

```

```
}

```

6. //spawn_meer

```
//Written by Garon Clements
```

```
//Script used to spawn meerkat while Ganesha is talking
```

```
void main()
```

```
{
```

```
    CreateObject(OBJECT_TYPE_CREATURE, "meerkat",
    GetLocation(GetWaypointByTag("Meer_Point")), FALSE, "Apart_Meer");
```

```
}
```

7. Added to on Item Acquire Script

```
// Addition by Garon Clements
```

```
// if the valid item's tag is "quinn_poem"
```

```
if (GetTag(oItem) == "quinn_poem")
```

```
{
```

```
// get the PC that acquired the item
```

```
object oPC = GetItemPossessor(oItem);
```

```
// on the PC, set the journal with the tag "jt_findNich"
```

```
// to its second entry
```

```
AddJournalQuestEntry("jt_findNich", 2, oPC);
```

```
}
```

