

BioWare Aurora Neverwinter Nights Toolset
Module Construction Tutorial

Plot Summary

A tribe of goblins led by Barash, a bugbear, is massing for war against the people of Hyatt, a small kingdom ruled by King Percy the Just.

The kingdom's primary source of iron for use in weapons is the Fernesk Mine outside the rural community of Fern. Every few weeks, a caravan of iron leaves Fern for Hyatt Castle to the South and returns with the goods necessary to maintain the town. The entire community depends upon the caravan for its very survival and they often hire mercenaries to serve as escorts.

Barash's first step is to secure the Fernesk Mine. This move serves two purposes: it cuts off the Castle Hyatt's supply of iron, and simultaneously increases his own. Barash has Gnashgab, one of his most trusted shamans, seize the mine, and capture the miners for use as slaves.

Under the cover of darkness, Gnashgab and her warriors invaded the mine while it was unoccupied and prepared an ambush for the next day's shift. The shift entered the mine normally the next day and fell victim to the goblins' trap. Now they work as slaves for the goblins, unable to escape and warn the rest of the town of the danger close at hand. Days have gone by since the last shift of workers entered the mine, and the time for the next caravan is fast approaching. The people of Fern are desperate.

Unknown to the townsfolk, Barash has an ally. Veran, a local merchant and the mine manager, is secretly working with the bugbear to provide information and supplies. Under the guise of forming a search party, Veran is really sending fresh slaves to Gnashgab to replace those that have died or been killed. Unfortunately for Veran, the people of Fern are too afraid to enter the mine.

Into this mess arrive the PCs, mercenaries hired to escort this latest caravan to Hyatt Castle. When the PCs enter Fern, they find that the caravan has been delayed indefinitely until the mystery behind the Fernesk mine is solved.

The mayor of Fern, Falstadd, asks the PCs to join Veran's search party to help find Falstadd's son, Jacen, who was part of the ill-fated work shift. If the PCs are unwilling to help, Falstadd offers them a reward.

During this adventure, the PCs discover a goblin-infested mine, the remains of the work shift slaving away and Jacen - one of the many victims of the attack.

Tutorial 1: Fern, the Frontier Town

Introduction

This tutorial introduces the New Module and New Area wizards and explains how to paint terrain.

Goal

Create the frontier town of Fern, the starting point for this module.

Tutorial

New Module Wizard

To create a brand new module, follow these steps:

1. Start the BioWare Aurora Neverwinter Nights Toolset. Select "Create a New Module" in the Welcome window and press "OK". If the Welcome window doesn't appear, select the "New" command from the File menu.
2. The New Module wizard will open. Press "Next" to begin.
3. Name the Module "tutorial" and click "Next".
4. Click "Area Wizard". Areas are where the module's action and story takes place. A module must have at least one area to work, but usually has several.
5. Name the area "Fern" and select the "Rural" tileset, then click "Next". Areas are constructed from 10-meter by 10-meter tiles. The tiles are grouped according to themes, which all tiles in the group share, such as City Exterior, Dungeon or Rural. These themes are called tilesets.
6. Accept the default area size of 8-tiles by 8-tiles. Press "Next" and then "Finish". The Toolset creates the area and returns to the New Module wizard.
7. Click "Next" and then "Finish" to exit the New Module wizard.
8. Save the module and exit the Toolset.

Take a look at the module in the game by following these steps:

1. Start the Neverwinter Nights game. On the main window, click "New".
2. Then press "Other Modules..."
3. Select the "tutorial" module and press "Load".
4. Pick a character (or create a new one).
5. Finally, press "Play".

After exploring the area for a few minutes, exit the game. Go back to the Toolset and load the module.

Camera Control

The Module Contents panel on the left side of the main window displays the areas, conversations and scripts contained within the loaded module. Expand the area list; the tutorial module only contains the one area, Fern. Double-click on the area's name or select "View Area" from its context menu to see it in the Area Display panel.

All areas will appear from a top-down perspective when they are first displayed. Notice the Camera Control Toolbar at the bottom of the area display panel. The first four buttons pan the camera around, the second four buttons rotate the camera and the next two

buttons control the zoom. The last two buttons allow you to control the orientation of game objects that have been painted in the area and will be discussed later.

The camera may also be controlled in these ways:

	Mouse	Numeric Keypad
Pan	[Ctrl] + left mouse drag	2 / 8 and 4 / 6
Rotate	1. [Ctrl] + right mouse drag. 2. Middle mouse drag.	1 / 3 and 7 / 9
Zoom	Wheel mouse	+ / -
Reorient		5

Table 1.1 Camera Control

Painting Terrain

The right side of the application window contains the Terrain and Game Object palettes. The palettes contain the feature, group and terrain tiles, and blueprints for the various game objects, which can be painted into the current area. There are several palettes, one for each type of game object. The blueprints for each palette are organized into categories to make finding a specific blueprint easier.

To paint some terrain:

1. The Terrain palette is on the left side of the top row of Palette buttons. Click the "Terrain" palette button to open it.
2. Expand the Terrain list. Select the "Raise / Lower Terrain" tool. Left click in the northeast corner of the area. Notice that a portion of the area has been raised up. Right click on the same spot and the raised section will be lowered back down.
3. Select the "Inn" from the Groups list. The inn is a tile group, which is a set of tiles that are painted together to form a larger structure. When the inn is selected, a ghosted image of it will appear, attached to the mouse. This image indicates the orientation the group will have when it is painted and can be changed by right clicking. Each right click changes the orientation of the ghosted tiles by 90 degrees. Paint the inn by left clicking in the area anywhere in the top right section of the map.

Note: The previous paint actions can be undone using [Ctrl] + Z or the "Undo" command from the Edit menu repeatedly. The previously undone paint actions can be reapplied using [Shift] + [Ctrl] + Z or the "Redo" command from the Edit menu.

4. Select one of the tiles that make up the inn. Notice that both tiles in the group are now selected. Group tiles cannot be selected independently. Remove the inn from the area by using the [Delete] key or painting over top one of the tiles with the Erase tool from the Terrain palette.
5. Repaint the inn, using [Ctrl] + Z, selecting the "Undo" command from the Edit menu or simply by selecting it from the Terrain palette and painting it again.

6. Select "Cave" from the Feature list on the Terrain palette. Paint the cave in the northeast corner of the area, changing its orientation so that it faces in toward the bottom of the area.

Note: For performance reasons, the Toolset does not continuously recalculate shadows. This means that while painting, shadow remnants and other lighting artifacts may be left behind. Press "F5" to recalculate shadows and other lighting effects.

Start Locations

When a player starts the module, the game will place their character at the module's start location. A module can have only one start location. By default, the start location is at the center of the first area created using the New Module wizard, although any area in the module will serve just as well.

To change this module's start location, follow these steps:

Click the Paint Start Location button on the far right of the top row of palette buttons. Paint it in a spot appropriate for the area, such as close to the inn.

Note: Start locations are a little different from terrain and game objects. There is no actual palette from which to select a start location. Once the button has been pressed, simply click in the area at the point where the characters should start.

1. Change the orientation at which the player's character will appear in the game by selecting the start location in the area and using the Rotate Object buttons on the Camera Control Toolbar or by using [Shift] + Right mouse drag.

Extra Credit

Continue to add other terrain features to the area. Perhaps a road runs through the town, or maybe the town sits on the edge of a lake or ocean. Remember to save the module regularly.

To view the module in the game at any time, follow the instructions given earlier.

Tip: To copy a set of connected tiles select the desired rectangular region and press [Ctrl] + C or use the "Copy" command from the Edit menu. To paste the region into the current area or another area (provided it uses the same tileset) press [Ctrl] + V or use the "Paste" command from the Edit menu. This will attach the tiles to the mouse in the same fashion as a tile group; just paint it in the area with the appropriate.

Proceed to the next tutorial when you have finished.

Tutorial 2: The Fernesk Mine

Introduction

This tutorial expands on the terrain painting skills learned in the previous tutorial and introduces new area manipulation commands. Painting and editing game objects is also covered.

Goal

Create an area for the Fernesk Mine and populate it with interesting objects.

Tutorial

A New Area

Create a new area to serve as the mines invaded by Gnashgab and her warriors by following these steps:

1. Start the Toolset and load the "tutorial" module.
2. From the Wizards menu, select "Area Wizard".
3. Name the area "Fernesk Mine", select the "Mines and Caverns" tileset and press "Next".
4. The mine area should be a little smaller, make it 5 tiles high by 4 tiles wide and click "Next".
5. Ensure that the "Open Area in the Area Viewer" checkbox is checked and press "Finish".

Tip: The Toolset uses the in-game lighting in the Area Display by default, but this may be too dark to work effectively. Several commands in the Environment menu enable and disable graphical features that will make it easier to see while you work on the area without affecting how it appears to the players. For example, to set lighting to very bright while working on a module, Click on Environment tab and toggle the Fog command and the Use Area Lighting command to off.

Note: Zoom out to get a view of the entire area. Notice that this area is mostly walled up and blocked off except for a small space in the center. This is because the initial terrain for this tileset is of type "Wall" and requires the playable area to be "carved" out of the surrounding environment.

6. Select "Wall" from the list of terrain types and fill in the region of the area that was initially populated with "Floor", so that the entire area is covered.
7. Select the "Exit 3" tile from the Feature list, orient it so that it faces north, into the area, and paint it in the bottom left corner (row 0, column 1).

Note: When an area is loaded, the default camera position is such that north is to the top of the screen.

Note: The status bar indicates mouse position (x, y), grid position (row, column) and the resource name of the tile on which the mouse rests.

8. Select "Tracks" from the list of Terrain and paint them up the left edge of the area, but do not run them off the end of the map.
9. Select "Mineshaft" from the Feature list and paint it at row 3, column 2 - close to the center of the area, creating an isolated room with the mineshaft in the center. This room will be connected to the tracks later.

Advanced Area Manipulation

As a result of the modifications to the area that introducing the "Mineshaft" caused, the area is now practically full. In its current state, adding new features or groups without altering what has already been created would be very difficult. Fortunately, the Toolset includes one very special command under the Edit menu: "Resize Area".

1. Select the "Resize Area" command from the Edit menu.
2. Enter the area's new dimensions, 8 tiles high by 5 tiles wide and press "OK".

WARNING: The Toolset adds to and removes from the top row and right column of the area when it is being resized. This normally isn't an issue if the area is enlarged but if the area is reduced, objects on the top rows and right columns tiles will be automatically deleted.

Tip: The "Resize Area" command can be undone using [Ctrl] + Z or the "Undo" command.

3. Select "Tracks" again and paint it up the left edge of the area the rest of the way, but don't run it off the end of the map.
4. Next, paint a little "Floor" towards the top center of the area, above the "Mineshaft" and to the right of the "Tracks".
5. Connect the "Mineshaft" with the "Tracks" using "Corridor". Do this again to connect the "Floor" with the "Tracks".
6. Finally, paint a little "Water" to the right and between the "Mineshaft" and "Floor".

Place the module's start location close to the exit, then save the module and close the Toolset. Load the module into the game and take a look around.

Game Objects

The palette on the right side of the window includes all the blueprints that you can paint in an area. A *blueprint* is a game resource, which represents an object that is placed in the game. When a blueprint is painted into the area, an instance of the blueprint is created at that location.

Note: The palettes are divided by blueprint type and are organized into appropriate categories. Each category will display either further subcategories or actual blueprint names. A category cannot have both subcategories and blueprints at the same level.

Tip: Use the "Object Preview" command, available from the View menu to examine the important details of a blueprint before you paint it into an area.

Place-able Objects

1. Start the Toolset and load the "tutorial" module.
2. Open the "Fernes Mine" area by double-clicking its name or selecting "View Area" from its context menu in the Module Contents panel.
3. Open the Place-able Object palette by pressing the fourth button from the right on the bottom row of the palette bar (it looks like a table).
4. Select "Table" from the "Miscellaneous Interior" category and paint it into the area, close to the exit.

Note: Once the table was painted, it appeared in the list of place-able objects beneath the area in the Module Contents panel. Every instance in the module is located under the appropriate type in the area where it was placed.

Note: Use the "Rotate Object" buttons on the Camera Control Toolbar or press [Shift] + right mouse drag to change the orientation of object instances.

5. Select "Chair" from the same category and place a few of them around the table.

Tip: To paint multiple copies of the same blueprint, hold the [Shift] key while you paint.

6. Paint a few crates from the "Containers and Switches" category around the entrance and by the mineshaft.

Tip: When a place-able object is initially painted down or selected, a wire-frame arrow will appear, pointing in the direction where a PC must stand in order to use it. Make sure that any place-able object that the player will have to interact with does not have its wire-frame arrow pointing into a wall or similar obstruction.

7. Paint a torch bracket from the "Miscellaneous" category close to the entrance. Set the torch bracket's initial state to "Activated" from its context menu. This ignites the torch.
8. At the end of the floor, towards the top right of the area, paint an Altar from the "Miscellaneous Interior" category and a couple of Braziers from the "Miscellaneous" category.
9. From the "Battlefield" category, add a couple of Arrow-Filled and Speared Corpses around the entrance, the mineshaft and on the floor at the top right.

10. Finally, place a Plague Victim Corpse on the altar. This corpse is Jacen, the son of Falstadd.

Tip: Maneuvering a place-able object that has been painted on top of another can be a little jarring because the object may fall through to the floor or ground and become partially obscured. Simply press F5 to refresh the area, and make all objects adjust themselves properly.

Check out the module in the game. Notice that some of the crates may have treasure in them. Certain objects have scripts automatically assigned to them that create treasure when they are opened.

Encounters

An *encounter* is a trigger that spawns creature instances from its list of specific creature blueprints according to the power of the creatures that activate it. The same encounter that spawns only one creature for a lone, low-level character may spawn four creatures for a group of low-level characters or even a single high-level character.

In general, it is better to use encounters for the typical fights that PCs engage in during the course of an adventure. This reduces the load on the server and takes into account the fact that the module designer may not know the exact makeup of the party in advance. Only creatures required to advance the story (such as plot-givers or villains) should be placed explicitly in the module.

1. Start the Toolset and load the "tutorial" module.
2. Open the "Fernes Mine" area.
3. Open the Encounter palette by pressing the third button from the left on the bottom row (it looks like a pair of crossed swords).
4. Select "Goblin Group" from the "Very Easy" category.
5. This encounter should be a rectangle and placed in front of the entrance to the Mine. Draw the region (polygon) where the PCs must walk in order to activate the encounter by placing the points where the edges meet. To close the polygon, double-click on the last point.

Note: Encounters are most effective when placed at choke points in the area, such as close to entrances, doorways and bridges. This ensures that the PCs will activate the encounter.

6. Select the encounter and select "Properties" from its context menu to open the Encounter Properties window. Change the difficulty from "Normal" to "Very Easy", set the minimum number of creatures to 2 and press "OK".
7. Select the encounter and right click just past the start of the tracks. Select "Add Spawn Point" from its context menu to place a spawn point for that encounter. A *spawn point* tells the encounter system to place the creatures at that location.

Note: Spawn points only appear in the display for the selected encounter. The size of the big arrow that indicates the position of the spawn point can be adjusted from the Options.

8. Place another very easy goblin encounter similar to the one described in step 4, before the entrance to the room with the mineshaft.

Check out the module in the game. This time, ensure your character is ready to fight!

Creatures

A *creature* is anything that moves. This includes PCs, NPCs and monsters.

1. Start the Toolset and load the "tutorial" module.
2. Open the "Fernes Mine" area.
3. Open the Creature palette by pressing the first button on the left on the bottom row (it looks like a monster head).
4. Select the first "Goblin Shaman" under the Monsters | Humanoid | Goblin category and place it in the room with the altar.
5. Right click on the goblin in the area and select "Properties" from its context menu. This opens the Creature Properties window.
6. Change the creature's first name to "Gnashgab".
7. On the Statistics tab, reduce the Base Hit Points to 10.
8. Go to the Spells tab and select the "Cleric" class. The cleric spell list should appear.
9. Select the spell "Aid" by clicking in its "Prepared" cell (press F1 to see the Spell Description).
10. To reduce the number of times Gnashgab has this spell memorized, change the value in the cell to 0 and press the "Enter" key. Notice that the spell summary, at the bottom of the tab, updates to reflect the change.
11. Press "OK" to accept the changes.

Extra Credit

Place an undead or vermin encounter towards the end of the tracks with its spawn point in the following room. In this case, leave the difficulty at Normal, but set the minimum number of creatures to 1.

Tip: Use the Edit | Find Instance command to locate a specific instance of a blueprint placed in any area in the module.

Note: Editing a blueprint does not impact any instances that were previously painted in any of the areas. Also, editing an instance of a game object in an area does not affect the blueprint from which it was painted.

Tutorial 3: Doors and Area Transitions

Introduction

In addition to introducing doors, this tutorial completes the terrain painting lessons by explaining how to link areas together.

Goal

The purpose of this tutorial is to link the area containing the town of Fern with the Fernes Mine area.

Tutorial

Doors

Doors are special kinds of game objects that only fit in specific places, called *hook points*, of certain tiles. In general, there are two kinds of doors, universal and tileset specific. Universal doors can be placed in any area. Tileset specific doors only fit in the current area because of its tileset.

1. Start the Toolset and load the "tutorial" module.
2. Open the Fern area.
3. Open the Door palette by pressing the second button from the left on the bottom row.
4. Select one of the universal wooden doors. Notice that a spinning door has been attached to your mouse cursor.
5. Bring the door close to the entrance to the Inn. The door should stop spinning and an image of the door should appear in the Inn's doorway.
6. Paint the door while the image is in the Inn's doorway, by left clicking in the area (the same way as any other object would be painted). The door instance should be selected.
7. Delete the door.
8. Select one of the tileset specific doors.

Note: A tileset-specific door will appear as a generic wooden door until it is painted into a hook point that expects a tileset-specific door.

9. Attempt to paint it in the Inn's doorway. The blueprint using, a generic door model, should appear.

Note: Painting a tileset specific door in a location that expects a universal door causes the door to appear with a generic appearance, although all its other properties are unaffected.

10. Delete the tileset specific door and repaint one of the universal ones.

Doors serve several purposes. The primary purpose is to block parts of the area off until the players meet certain conditions, like retrieving a key. In addition, a door may serve as an area transition, a link between areas.

Area Transitions

An *area transition* is the means by which players are transported from one area to another. Transitions can originate from a door or trigger and can link to a corresponding door, trigger or waypoint.

1. Open the Fernesk Mine area.
2. Select the door in the entrance.
3. Select "Properties" from its context menu.
4. Change the door's script tag to "dt_MineToFern" and press "OK".

Note: Area transitions set up this way rely on the script tag to work properly. Each door should have a unique script tag; otherwise the in-game results are unpredictable. The Toolset displays a warning if an attempt is made to link an area transition to a door whose tag has not been changed.

Tip: Try starting the script tag of each object with the first letter of the object type, followed by a "t_" and then a name, such as "MineToFern", which, in this case, indicates the areas and direction of travel involved in an area transition.

5. Open the Fern area, saving your changes when prompted.
6. Select the "Cave Entrance" door and open its properties.
7. Change the door's script tag to "dt_FernToMine".
8. On the Area Transition tab, click "Setup Area Transition" to open the Area Transition window.
9. Select "Fernesk Mine" from the Target Area drop down. Leave the Target Type and Connection Type values alone.
10. Select "dt_MineToFern" from the list of Available Doors, press "OK" to close the Area Transition window and press "OK" to save the changes to the area.
11. Press "OK" to close the Door Properties window.
12. Place the module's start location close to the Inn.

Load Screens

Each area has a default load screen associated with it, based upon its tileset, which appears when players transition to that area. This load screen can be specified from the Area Properties window. Area transitions may also have a load screen attached, which overrides the default one. By default, the game uses a random load screen appropriate to the tileset used for the target area.

1. Start the Toolset and load the "tutorial" module.
2. Open the Fern area.
3. Select "Area Properties" from the Edit menu.

4. Go to the "Advanced" tab, and press the button next to the named Loading Screen, which should initially say "Random", to open the Loading Screen Selection window.
5. Uncheck the "Use Random Loading Screen" checkbox.
6. Select the desired loading screen from those available.
7. Switch to the Fernesks Mine area.
8. Select "Properties" from the context menu of the door in the cave entrance, the one with the Script Tag: dt_MineToFern.
9. Switch to the "Area Transition" tab, and press the button next to the named Loading Screen, which should say "Default" initially. This means that it will default to the one specified in the area's properties.
10. Uncheck the "Use Random Loading Screen" checkbox.
11. Change the filter to use the "Rural" tileset and select a different load screen than the one used as the default in step 6.

Save the changes to the module and check it out in the game.

Extra Credit

Create an interior area, based on the "City Interior" tileset, and link it to the Inn in Fern using doors.

Lock the door to the Inn, by setting the door's Locked property. On the same tab, specify the script tag of the item required to unlock the door. This allows any creature that has an item with this tag in its inventory to open the door.

Tip: It is possible to set up an "Area Transition" within the same area.

Play around with tileset-specific and universal doors. Find a place to paint a "Wall 2 Gate" terrain feature. Select and delete its door. Select a tileset-specific door and paint it in that feature, delete it and then do the same with a universal door. In both cases, the same door model should appear, but there should be subtle differences in the door properties reflecting that different blueprints were being used.

Tutorial 4: Create the Cast

Introduction

Now that the two areas, Fern and the Fernesks Mine have been created and linked, the players can travel between them freely. The next step is to create the major players in this adventure. These are the characters that have a significant role to play in the story, whether it is through combat or conversation.

This tutorial describes the difference between the standard and custom palettes and demonstrates the different ways to create custom blueprints

Goal

The purpose of this tutorial is to create custom creature blueprints for Falstadd, Gnashgab and Veran, each using a different method.

Tutorial

Standard vs. Custom Palettes

Each type of game object has a palette where the different blueprints are listed. Each palette is divided into two sub-palettes: Standard and Custom. The standard palette contains the blueprints that are available to every module. The custom palette contains those blueprints that are only available in the current module.

There are several ways to add unique blueprints to the custom palette.

Gnashgab

Gnashgab is the goblin shaman who led her warriors into the Fernes Mine and captured the workers, turning them into slaves.

1. Start the Toolset and load the "tutorial" module.
2. Open the Fernes Mine area.
3. Find and select Gnashgab in the area's list of creatures from the Module Contents panel.
4. Select "Focus on Object" from Gnashgab's context menu. This should change the area display so that Gnashgab is in the center and surrounded by a bounding box.
5. Right click on Gnashgab to bring up her context menu and select "Add to Palette". This activates the Creature Properties window.
6. A unique blueprint name will be generated automatically, so all that remains is to specify a new palette category, Special | Custom 1, on the "Basic" tab.

Note: Neverwinter Nights requires that each game object blueprint have a unique name. This name is neither the object's name that appears in game, nor is it the object's script tag, it is a separate name used to uniquely identify the blueprint. In most cases a unique blueprint name is automatically generated. If required, one can also be specified manually with the following restrictions:

1. No two blueprints can have the same blueprint name.
 2. A blueprint name can be no more than 16 characters long.
 3. A blueprint name can only contain the letters a - z (lowercase only), numbers 0 - 9 and the underscore "_" (without the quotes).
7. Press "OK". This creates a new blueprint for the goblin Gnashgab, allowing her to be placed at other points in the module, or to be spawned in on demand by the DM during play.

Falstadd

Falstadd is the mayor of Fern, and the father of Jacen, one of the workers who were captured in the raid.

1. Switch to the Standard creature palette, if it isn't already selected.
2. Select "Shop Keeper" from the NPCs | Humans category.
3. Select "Edit Copy" from the Shop Keeper's context menu. This will open the Creature Properties window.
4. Change the creature's first name to Falstadd and give him a random second name by clicking the "Generate Random Name" button - it has a picture of a face on it.
5. Change the Script Tag to "ct_Falstadd" (without the quotes, of course).
6. Change the blueprint's category to Special | Custom 2.
7. Press "OK". Notice that the creature palette immediately switches to the custom sub-palette and selects the new blueprint.
8. Place an instance of Falstadd near the player's start location in the Fern area.

Veran

Veran is the collaborator, working with the goblins to enslave the workers and conquer the kingdom of Hyatt.

1. Select "Creature Wizard" from the Wizards menu. This starts the New Creature Blueprint wizard. Press "Next" to begin.
2. Set the Racial Type to "Human" and press "Next".
3. Remove the "Fighter" class by pressing the red "x" next to the level.
4. Add the "Rogue" class and set the level to 3. Press "Next" to continue.
5. Leave Veran's gender alone, but select the "Human, Male 01" appearance and choose a portrait.

WARNING: All creatures, doors and place-able objects require a portrait.

6. Press "Next", leave Veran's faction as "Hostile" and press "Next".
7. Enter "Veran" as the creature's first name and press "Next".
8. Select the Special | Custom 1 category for this blueprint.

Note: All the villains have been going into the Special | Custom 1 category. It isn't required, of course, but organizing the palettes along plot lines does help the DM to locate special NPCs quickly.

9. Press "Next".
10. Review the summary describing the creature blueprint about to be created and press "Next".
11. Finally, ensure that the "Launch Creature Properties" checkbox is not checked and press "Finish".
12. Place an instance of Veran a couple of tiles or so away from the player's start location and the instance of Falstadd.

Save the module, exit the Toolset and check it out in the game. Veran will probably attack either Falstadd or the PC immediately, depending upon which is closer, because of the Faction system and the default scripts assigned.

Tip: To delete a blueprint, simply select "Delete" from its context menu. Standard blueprints cannot be deleted.

Extra Credit

D&D Rules Compliance in the Toolset

Using the Creature Wizard, it is possible to create creatures in accordance, approximately, with the 3rd Edition D&D rules. Using the Creature Properties window, it is possible to make creatures that deviate significantly from those rules.

The "Special Abilities" tab in the Creature Properties window allows fantastic abilities, abilities outside the scope of any assigned class, to be given to a creature.

1. Create a new Elf blueprint with 1 level of Wizard using the Creature Wizard.
2. Ensure that the "Launch Creature Properties" checkbox is checked before finishing the wizard.
3. The Creature Wizard automatically assigns an appropriate number of spells to the creature (assuming no bonus spells for intelligence). Try adding 2 or more level 0 (cantrip) or level 1 spells or any number of higher-level spells and pressing "OK". The Toolset should present a warning indicating that the creature has been assigned too many spells for its current level or that it cannot cast spells of that level.
4. Press "No" to return to the Creature Properties window and remove the offending spell(s).
5. Switch to the "Special Abilities" tab, assign the same spell that was assigned in step 3 and press "OK". The Toolset should accept this and allow the properties window to close.
6. Delete this blueprint.

Note: Special and mundane abilities of items in the inventory of a creature are not reflected on its Statistics tab.

Tutorial 5: General Attitudes: Factions

Introduction

At the end of the last tutorial, the unsuspecting townspeople of Fern were left at the mercy of Veran, a homicidal maniac. This makes it a little too easy for the players to identify the character as a possible villain, however. It is necessary to make Veran a little more tolerant of other creatures.

Goal

This tutorial explains Factions and introduces the Faction Editor in the Toolset. In addition, two new factions and a new creature blueprint will be created.

Tutorial

Factions

Factions are a means of describing the general relationships between groups of creatures and other objects, including the PCs. The default scripts assigned to the events of various game objects rely on this system to determine the expected behavior of the object in its environment. This is why Veran brutally attacked anyone that did not belong to his faction.

Each module starts with the five standard factions: PC, Hostile, Commoner, Merchant, and Defender. Game objects can be assigned to any faction except PC. New factions may also be created using the Toolset, which allows custom alliances and hatreds to be crafted without having to get into the scripting language.

Collaborator Faction

Follow these steps to assign Veran to the Collaborator faction. The Collaborator faction is similar to Merchant, but isn't as adversarial towards the Hostile faction.

1. Start the Toolset and load the "tutorial" module.
2. Select "Faction Editor" from the Tools menu.

Tip: The Faction Editor displays information in a bar graph on the "Basic" tab and in a grid on the "Advanced" tab.

3. Press "Add Faction" in the lower right of the window.
4. Name the faction "Collaborator" and set its parent faction to "Merchant". Leave the Global Effect checkbox alone and press "OK".
5. A new faction should appear in the list on the right.

Note: The checkboxes beside each entry in the list on the right indicate that the faction appears in the graph or table.

Note: The value of the column in the graph, or cell in the grid shows how the selected faction in the list feels about the checked factions.

	Checked Faction		
Value	0 - 10	11 - 89	90 - 100
Selected Faction	Enemy: Attack on sight.	Neutral: Ignore.	Ally: Will aid if attacked.

Table 5 - 1: Faction Values

6. With everything else checked, select the Hostile faction from the list. Notice that in the graph it shows that members of the Hostile faction consider members of every other faction an enemy.
7. With the Hostile faction still selected, grab the bar for the Collaborator faction in the graph and drag it up until it is set to 50. This means that members of the Hostile faction no longer automatically attack members of the Collaborator faction.

Tip: If dragging the bar is too awkward, use the grid on the other tab to set a specific value.

Note: The PC faction cannot be adjusted.

8. Select the Collaborator faction from the list. Drag the Hostile faction bar in the graph until it is set to 50. This means that members of the collaborator faction no longer automatically attack members of the Hostile faction.
9. Press "OK". It may take a few moments for the Toolset to save the new faction information.

Veran

Assign Veran to this new faction.

1. Open the Fern area.
2. From the custom Creature palette, locate the blueprint for Veran and select "Edit" from its context menu.
3. On the "Advanced" tab, set the Faction to Collaborator.
4. It is important to remember that modifications to the blueprint are NOT automatically reflected in any instances of that blueprint that have already been placed in the module. Press "Update Instances", then press "Yes" to update all instances in the module, press "No" to avoid seeing a list of all the instances that are about to be updated, wait for a few moments, and finally press "OK".
5. Press "OK" to close the Creature Properties window.
6. Find the instance of Veran in the area display or from the module contents list and edit the instance's properties to verify that the faction has been set correctly and press "OK".

Save the module, exit the Toolset and load the game to see the changes. Veran should be much calmer now.

Miner Faction

What's a mine without miners? The former miners - now slaves of the goblins - should also be present in the Fernesk Mine area; so another faction should be created for them.

1. Start the Toolset and load the "tutorial" module.
2. Select "Faction Editor" from the Tools menu.
3. Add a faction, Miner, with Commoner as its parent.

Tip: If the graph is getting a little too cluttered, uncheck the Merchant and Defender factions in the list.

4. Select the Hostile faction. Set its value for the Miner faction to 15 by clicking on the bar and dragging.
5. Select the Miner faction. Set its value for the Hostile faction to 15 by clicking on the bar and dragging.

Note: Ratings between factions do not have to be reciprocal. That is, just because members of faction A consider members of faction B to be allies, members of faction B are not required to consider members of faction A allies.

6. Press "OK" to close the Faction Editor and accept the changes.

Miner

The next step is to create a new creature blueprint for miners.

1. Select "Creature Wizard" from the Wizards menu.
2. Create a human, with 1 level of commoner, a portrait (pick one) and the beggar appearance.
3. On the Faction tab of the Creature Wizard, select Miner, and press "Next".
4. Change the first name to Miner, blank out the last name and press "Next".
5. Assign the creature blueprint to the Special | Custom 2 category and finish the wizard.
6. Open the Creature Properties window for the blueprint by selecting "Edit" from the context menu for the blueprint from the palette. Change the Script Tag to "ct_Miner" and press "OK".

Miner Group Encounter

Finally, a new encounter should be created that spawns these creatures as the PCs explore the Fernesck Mine.

1. Select "Encounter Wizard" from the Wizards menu.
2. Specify the Special | Custom 2 category and press "Next".
3. Select the Miner blueprint from the custom creature palette, add it to the encounter's creature list by pressing the "->" button and press "Next".
4. Name the new encounter blueprint "Miner Group", ensure that the "Launch Properties Dialog" checkbox is not checked and press "Finish".
5. Open the Fernesck Mine area and paint an instance of the Miner Group encounter close to the second goblin encounter, just outside the room with the mineshaft.
6. Place a spawn point for the encounter close to the mineshaft itself.

Save the module and check it out in the game.

Extra Credit

The Waypoint palette includes blueprints for map notes, posts and waypoints. Map notes will appear in the player map and should be used to indicate the location of special places in the area. Place a map note for the entrance to the mine - its name will have to be changed.

The default scripts assigned to creatures use waypoints and posts to control their movement. Check the Comments on each to see how to use posts.

Use waypoints to give the miners the appearance that they are doing work. Place a few waypoints around the mineshaft. Group select them, and select "Create Set" from the context menu. Specify "WP_ct_Miner" as the name of the set. This automatically names the waypoints so that the miners will walk between them.

Note: None of the map note, post or waypoint instances will appear in the game as they do in the Toolset.

Tutorial 6: Journals and Conversations

Introduction

Currently, the tutorial module contains a couple of areas complete with instances of some interesting blueprints, a few unique NPC blueprints and a couple of new factions. It still lacks the hooks required to interest the PCs and encourage them to start the adventure, however.

This tutorial introduces the Journal and Conversation Editors.

In general, the state of a plot or quest is measured by its journal entries but is driven by the dialogue spoken by its participants.

Goal

Create conversations for Falstadd and Veran in which the PCs can participate, and which modify their Journal.

Tutorial

Quests and Journals

Adventures revolve around the quests that the PCs embark upon. The in-game journal allows the players to keep track of each step in the quest, and quickly see where to go next. Each quest is reflected by a journal category. The steps in the quest are entries in the category.

1. Start the Toolset and load the "tutorial" module.
2. Select "Journal Editor" from the Tools menu.

3. The buttons on the left side include Add, Cut, Copy and Paste. Press the "Add" button; only the "Add" button is enabled initially. This creates a new journal category.
4. Select the new category. Change the category's name to "Falstadd's Quest", its Script Tag to "jt_Falstadd", its Priority to "Highest" and its XP Value to 2000.
5. Add three entries to the category by pressing "Add" or selecting "Add" from the context menu while the category node is selected. Each entry represents a particular state in the quest.
6. Select the first entry and change its text to "Find Falstadd's son, Jacen, in the Fernesk Mine."
7. Select the second entry and change its text to "Return Jacen's ring to his father, Falstadd."
8. Select the third entry and change its text to "Jacen's ring has been returned to Falstadd." Check the "Finish Category" checkbox and press "OK".

Conversation: Falstadd

Conversations are an alternating sequence of NPC statements (or questions) and PC responses.

1. Select the Falstadd blueprint from the custom creature palette and select "Edit" from its context menu.
2. On the "Basic" tab, press "Edit" next to the Conversation drop-down at the bottom of the window. This launches the Conversation Editor and creates a new, empty conversation file. Notice how closely the Conversation Editor resembles the Journal Editor.
3. Press "Add" to create a new NPC statement and after entering the text below into the window that appears, press "OK". Notice that the text now appears in the tree, below the Root.

Node 1	NPC Text	
		Hello again, <FirstName>. Have you learned anything about the mine?

Note: <FirstName> is a token. A token is a special string that the game understands and replaces with some other text appropriate in the current context. In this case, the game would replace <FirstName> with the first name of the player to whom the NPC is speaking. Select "Insert Token" from the context menu in the text box to see a list of tokens and some examples.

4. With Node 1 ("Hello again...") selected, press "Add". This creates a new PC response node. Enter this text:

Node 2	PC Text	Yes, the mine had been invaded by some goblins and they were using the miners as slave laborers. The goblins were defeated, but unfortunately, Jacen was one of their victims. This ring was found on his body.
---------------	----------------	---

5. With Node 2 ("Yes, the mine...") selected, press "Add". This creates another NPC node. Enter this text:

Node 3	NPC Text	Thank you for your service to Fern, but... my poor son!
---------------	-----------------	---

6. With Node 1 ("Hello again...") selected, press "Add" again and enter this text as an alternative response for the PC.

Node 4	PC Text	No, not yet.
---------------	----------------	--------------

7. With Node 4 ("No, not...") selected, press "Add" again and enter this text for the new NPC node.

Node 5	NPC Text	You should probably speak with Veran over by his shop. He's looking for people to form an expedition into the mine. I would be very grateful if you would help him locate my son.
---------------	-----------------	---

8. With the Root selected, press "Add" and enter this text as an alternative greeting by the NPC.

Node 6	NPC Text	Welcome! I am Falstadd, mayor of Fern. Is there anything I can do for you?
---------------	-----------------	--

9. With Node 6 ("Welcome! I am...") selected, press "Add" and enter this text as the PC response.

Node 7	PC Text	Hello, I am <FullName>. I understand you need escorts for an iron shipment that is leaving for Hyatt Castle in the morning.
---------------	----------------	---

10. With Node 7 ("Hello, I am...") selected, press "Add" and enter this text for the next NPC statement.

Node 8	NPC Text	Yes, the caravan. You see, the shipment is not ready yet. The miners, including my son, Jacen, seem to have disappeared.
---------------	-----------------	--

11. With Node 8 ("Yes, the caravan...") selected, press "Add" but this time, enter no text.

Node 9	PC Text	
---------------	----------------	--

Note: A PC node with no text appears as either a [CONTINUE], or [END DIALOG] node in the game depending on whether another NPC node follows it. If a conversation ends with an NPC node, the game automatically adds an [END DIALOG] node.

12. This next node should tell the users to find Veran and get more information from him. A node like this has already been created, Node 5 ("You should probably..."). It is possible to create a link from the last node, Node 9 to Node 5. Select "Copy" from the context menu of Node 5. Select "Paste as Link" from Node 9's context menu. A grayed version of the text in Node 5 should now appear below Node 9.

Note: Use links to allow players to follow different paths in a conversation while ensuring that they are being provided with the same information. This is especially important when dealing with any plot information.

13. Save the conversation under the name "falstadd", and close the Conversation Editor. Save the changes to the Falstadd blueprint and either update all instances of it or simply delete the one instance that has been placed and repaint it.

Save the module and check it out in the game by talking with Falstadd.

Script Wizard

It should be clear that the conversation still requires some work before it makes sense. Fortunately, it won't require a lot of work - in fact only four carefully placed scripts, each of which is created using the Script Wizard.

1. Start the Toolset and load the "tutorial" module.
2. From the module contents panel, open the list of conversations and select "Edit" from the context menu for the conversation named "falstadd". This opens the conversation without having to navigate through the creature properties.
3. Select Node 1 ("Hello again..."). On the "Text Appears When..." tab, press the Script Wizard button (it looks like a wizard's hat). Check the "Local Variable" checkbox and press "Next".
4. Don't freak out. It isn't that bad.

Note: The use of variables requires a little planning, but they aren't that hard. A variable requires a name and represents a certain kind of data (like a number or set of characters - called a string). Set the value of a variable in one location based on some action or event and check it in another to confirm that the action or event took place.

5. Enter "nFirstTimeTalked" in the top right field. This is the variable name; it is set using another script.
6. Enter "1" in the bottom right field (next to the "Add" button). This is the value the variable must have before the script will return TRUE.

7. Press "Add" to add this check to the script and then press "Next".

Note: You can add more variables by continuing to name variables, set conditions (is equal to, is less than, etc.) and specifying values to compare them against.

8. Give the script the name "sc_falstadd01" and press "Finish".
9. Select Node 7 ("Welcome, I am..."). On the "Action Taken" tab, press the Script Wizard button. Check the "Set Local Variables" checkbox and press "Next".
10. Enter "nFirstTimeTalked" in the top right field.
11. Enter "1" in the field below the variable name and press "Add".
12. Press "Next", give the script the name "at_falstadd01" and press "Finish".
13. Select Node 2 ("Yes, the mine..."). This node should only appear once the players have investigated the mine and found Jacen's ring.
14. On the "Text Appears When" tab, press the Script Wizard button. Check the "Item in Inventory" checkbox and press "Next".
15. Enter the script tag of Jacen's ring, "it_RingJacen". This ring will be created and placed a little later.
16. Press "Add", then "Next". Give the script the name "sc_falstadd02" and press "Finish".
17. Select Node 2 ("Yes, the mine..."). Once this node is selected, the player should return the ring to Falstadd, which means it needs to be removed from the player's inventory.
18. On the "Action Taken..." tab, press the Script Wizard button. Check the "Take from Player" checkbox and press "Next".
19. Enter the Script Tag of the item to remove, "it_RingJacen" in this case, and press "Add".
20. Select Keep, below the list of required items. This means if the creature were to die, this item would be left behind in its inventory.
21. Press "Next", name the script "at_falstadd02" and press "Finish".

Save the conversation and the module, then exit the Toolset and check out the changes by talking with Falstadd twice.

Attaching Journal Entries

The PC's journal should be updated at certain points in the conversation.

1. Start the Toolset and load the "tutorial" module.
2. Open the falstadd conversation from the list in the module contents panel.
3. Select Node 5 ("You should probably..."). On the "Other Actions" tab, select "Falstadd's Quest" and "1" as the journal category and entry, respectively.
4. Select Node 3 ("Thank you for..."). On the "Other Actions" tab, set the journal category and entry to "Falstadd's Quest" and "3" respectively.

Again, check out the changes to the module in the game. Notice the journal is updated after talking with Falstadd.

Conversation: Veran

When, after speaking with Falstadd, the PCs come to talk to Veran, he's going to need something to say.

1. Start the Toolset and load the "tutorial" module.
2. Select the Veran blueprint from the custom creature palette and select "Edit" from its context menu.
3. Press "Edit" next to the conversation field at the bottom of the "Basic" tab to open the Conversation Editor with a new conversation.
4. Select the Root node, press "Add", and enter the following text.

Node 1	NPC Text	Greetings! I am Veran. I manage the local iron mine and own Veran's Mercantile. You'll not find better prices anywhere, I might add.
---------------	-----------------	--

5. Select Node 1 ("Greetings! I am..."), press "Add" and enter the following text.

Node 2	PC Text	Hello, Veran. I am <FirstName>. Falstadd suggested that you could use some help with an expedition to locate the missing miners.
---------------	----------------	--

6. Select Node 2 ("Hello, Veran..."), press "Add" and enter the following text.

Node 3	NPC Text	Ah, poor Falstadd. His own son, Jacen, is one of the missing miners. If you're willing to help, I can offer you a great deal on anything in the store. Would you like to see what I have for sale?
---------------	-----------------	--

7. Select Node 3 ("Ah, poor..."), press "Add" and enter the following text.

Node 4	PC Text	Yes, please.
---------------	----------------	--------------

Note: Merchants are explained in the next tutorial.

8. Select Node 3 ("Ah, poor..."), press "Add" and enter the following text.

Node 5	PC Text	No, thank you.
---------------	----------------	----------------

9. Save the conversation and the changes to the Veran blueprint. Delete the instance of Veran placed previously in the Fern area and paint another one or use the "Update Instances" command from the blueprint's context menu on the custom creature palette.

Extra Credit

Place an instance of the standard Innkeeper blueprint from the NPCs | Humans category in the in the Inn area, created as part of the Extra Credit step in Tutorial 3. If the Inn area hasn't been created, simply place the Innkeeper instance in the Fern area, outside the Inn.

Write a conversation that awards the PC with 1,000 experience points (XP), enough for a 1st level character to reach level 2, and assign it to the instance of the Innkeeper.

Tutorial 7: Items and Inventory

Introduction

Veran's conversation from the last tutorial was written such that it offered the player the opportunity to purchase some goods from a store. Goods can be exchanged between a player and an NPC through an instance of a Merchant game object. It is a good idea to allow the players to equip themselves before heading off on the adventure. The items can be mundane or magical and just like everything else, if the available items are unacceptable, they can be modified and even custom ones can be created.

This tutorial discusses item creation and merchants.

Goal

The goals of this tutorial are to place Jacen's ring that is found on his body in the mine, create a magical amulet to be assigned to a merchant instance, and to trigger that merchant instance through Veran's conversation.

Tutorial

Jacen's Ring

The first thing to do is place an instance of an existing item, a ring, in the inventory of the corpse on the altar.

1. Start the Toolset and load the "tutorial" module.
2. Open the Fernesk Mine area.
3. Select "Properties" from the context menu for the Plague Victim Corpse that was placed on the altar to open the instance's Place-able Object Properties window.
4. Check the "Plot", "Usable" and "Has Inventory" checkboxes.
5. Press "Inventory" to open the object's Inventory Properties window.
6. Select "Copper Ring" from the Miscellaneous | Jewelry | Rings category on the Standard palette and drag it over to the "Contents" tab.

Note: To remove an item from the inventory, drag it from the "Contents" tab to the garbage can icon at the bottom of the window.

7. Right click on the ring in the "Contents" tab to open the Item Properties window for the ring instance. Change the Script Tag to "it_RingJacen".

Note: This must be the same script tag used in the previous tutorial to control whether Node 2 in Falstadd's conversation appears.

8. Press "OK" to close the Item Properties window.

9. Press "OK" to close the Inventory Properties window.
10. Press "OK" to close the Place-able Object Properties window.

Amulet of Protection +1

Next, a new blueprint for an amulet of protection will be created. An instance of this amulet will be placed in the store for the PCs to purchase at the start of the adventure.

1. Select "Item Wizard" from the Wizards menu.
2. Select the "Amulet" item type and press "Next".
3. Enter the amulet's name, "Whyte's Amulet", leave the "Magical" checkbox unchecked and press "Next".
4. Assign the amulet to the Special | Custom 1 category and press "Next".
5. Check the "Launch Item Properties" checkbox and press "Finish".
6. Once the Item Properties window appears, switch to the "Properties" tab.
7. Find the "AC Bonus" property in the list on the left side of the tab and assign it to the amulet by pressing the "->" button.
8. Check the "Identified" checkbox.
9. Switch to the "Description" tab, and enter some details about this magical amulet.

Tip: Unidentified descriptions appear when the item is not identified. Identified descriptions appear after the item has been identified. You can use them both to provide clues to the players about the item they have acquired.

10. Press "OK" to close the Item Properties window. Notice that the amulet appears in the custom item palette.

Managing Merchants

Merchants are a little unusual. Merchants look like waypoints, but they have inventories like creatures and place-able objects.

1. Open the Fern area in the area display.
2. Open the standard Merchant palette (the 5th button from the left on the bottom row - it looks like a stack of coins).
3. Select the first "General Store" blueprint and place an instance of it close to the instance of Veran.

Note: It doesn't matter where the instance of the merchant is placed because players do not interact with the instance directly. Instead, players activate a merchant instance through a creature, door, place-able object or trigger. For organizational purposes, it's usually a good idea to put them close to the object the player interacts with to activate it.

4. Select "Properties" from the merchant instance's context menu to open the Merchant Properties window.
5. Change the Script Tag to "mt_Veran".

6. Adjust the Sell Markup field to 1. That is, the merchant will sell items for 1% of its actual cost.

Note: Generally, merchants purchase items for a fraction of their cost (< 100) and sell items for a profit (> 100). In this case, the merchant is both buying and selling items for 1% of their actual cost.

WARNING: Merchants will never operate at a loss. That is, the Sell Mark Up must always exceed the Buy Mark Down.

7. Press "Inventory" to open the Inventory Properties window for the merchant instance.
8. Find Whyte's Amulet, created earlier, from the custom item palette and drag it into any of the contents tabs. The Toolset automatically adds an instance of the item blueprint to the merchant's inventory and selects it (displaying the tab on which it can be found).

Note: The Options section on the Inventory Properties window indicates the merchants buy and sell prices (how much the merchant pays for and charges for the item). With a sell markup of 1%, Whyte's Amulet, with a cost of around 600gp, would only cost 6gp.

9. Press "OK" to close the Inventory Properties window.
10. Press "OK" to close the Merchant Properties window.

Opening a Store

Finally, the merchant has to be activated based on some action taken by the player. It is easiest to simply hook it up to a conversation.

1. Open Veran's conversation from the list of conversations in the module contents list by selecting "Edit" from its context menu.
2. Select Node 4 ("Yes, please."). On the "Actions Taken..." tab, press the Script Wizard button.
3. Check the "Perform an Action" checkbox and press "Next".
4. Select the "Start a Merchant" radio button and enter the tag: "mt_Veran".
5. Press "Next", enter the name of the script "at_veran01", and press "Finish".

Save the conversation, close the Conversation Editor, save the module and check it out in the game.

Extra Credit

Edit the instance of Veran and move the dagger from his right hand to his inventory.

Create a Book item with a couple of details about "The Plan" and the existence of an ally in the town. Place this note on Gnashgab for the PCs to find. After placing this object in Gnashgab's inventory, select it and set the Droppable flag (check the checkbox).

Tutorial 8: Atmosphere

Introduction

With the clever use of lighting and sound, an area that the players have visited before and are familiar with can be transformed into something completely new and unrecognizable.

This tutorial introduces ways to manipulate area lighting and sound settings.

Goal

At the end of this tutorial, the Fern and Fernesk Mine areas will be lit properly and include appropriate sounds and sound objects.

Tutorial

Area Lighting: Fern

This part of the tutorial will demonstrate how to adjust lighting on an area-wide scale.

1. Start the Toolset and load the "tutorial" module.
2. Open the Fern area. Ensure that the area display is using the area lighting by checking the Environment menu or the toolbar.
3. From the area list in the module contents panel, select "Properties" from the Fern area's context menu or "Area Properties" from the Edit menu.
4. Switch to the "Visual" tab. Click on the "Exterior, Sunset" setting and press "Apply". Notice how the area lighting changes to a nice orange.
5. This is not quite ominous enough; the main-sun light should be toned down. Press "Customize Environment" to open the Environment Options window.
6. Change the Sun Diffuse Color by clicking on the color box. This opens the Color window. Press "Define Custom Color" and enter the following values for Red: 100, Green: 150, Blue: 160.
7. Press "OK" to close the Color window, press "OK" to close the Customize Environment window and finally press "OK" to close the Area Properties window.
8. To see the full effect of the lighting scheme, it is important to enable fog for rendering. By default, fog is not rendered in the Toolset. Select "Fog" from the Environment menu.

Note: Depending upon how far the camera is from the ground, the area may be partially or completely obscured by the fog. If necessary, zoom in to see things a little better.

9. The last step is to add some weather effects. Open the properties for the Fern area once more. Switch to the "Visual" tab, and press "Customize Environment".
10. Set the Weather - % Rain field to 100. This means there is a 100% change of rain (it is always raining) when the players enter this area.

Note: Weather effects are not visible in the Toolset.

11. Press "OK" to close the Customize Environment window, and then press "OK" to close the Area Properties window.

Save the module and check it out in the game.

Tile Lighting: Fernesk Mine

Area-wide lighting provides a means to set the general feel for the area. There are times, however, when it is important to draw the player's attention to specific parts of an area. This is where tile lighting is useful.

1. Start the Toolset and load the "tutorial" module.
2. Open the Fernesk Mine area in the area display. The Toolset should still use the area lighting; check the Environment menu or toolbar if necessary.
3. Open the Area Properties window and switch to the "Visual" tab. Select "Interior Dark" and press "OK". The area should be a lot darker now.
4. Center the camera on Gnashgab by selecting the creature from the list in the module contents panel and selecting "Focus on Object" from its context menu.
5. Manually switch to the area selection mode to terrain by selecting "Select Terrain" from the View | Selection Mode menu.
6. Select "Tile Properties" from the context menu of the tile where Gnashgab and the other objects are located. If they are on multiple tiles, simply repeat the next step for all the tiles involved.
7. The Tile Properties window should now be open. Click on the color assigned to "Main Light 1" to open the Select Color window. Select orange (the bottom right color) and press "OK".

Tip: The RGB value appears in the tool-tip for the color.

8. Press "OK" to close the Tile Properties window.
9. Find the room with the mineshaft. Turn off the use of area lighting, if necessary, until you locate the right spot.
10. Turn on area lighting, select the tile with the mineshaft and change its "Main Light 1" to gray (top row, third one in from the left).
11. Finally, edit the properties of the Exit tile, where the PCs enter the mine. Set the "Main Light 1" to a soft green (second row, 3rd one in from the left).

Note: changing the overall environment overrides any modifications made to the properties of any specific tiles.

Save the module again and check it out in the game.

Ambient Sounds / Music: Fern

Ambient sounds serve a similar purpose as the area lighting; they help set the mood and tone of the area.

1. Start the Toolset and load the "tutorial" module.
2. Open the Area Properties for the Fern area.
3. Select the "Audio" tab. In keeping with the rural tileset used for Fern, set the Ambient Sound Day field to "Town, Day - Sparse" and the Ambient Sound Night field to "Town, Night".

Note: The important thing about the main area ambient track is that it helps set the mood. You can take the same area, one with a spooky forest with creaking branches, and one with a pleasant sound with a light wind in the leaves, and create two completely different settings.

4. The "Audio" tab on the Area Properties window also has options for the day, night, and battle music tracks. Leave these alone for now, as the defaults are sufficient.
5. Press "OK" to close the Area Properties window.

Tip: To hear the sound and music in the Toolset, check the "Play Ambient Sound" and "Play Ambient Music" options in the Environment menu or the press the corresponding buttons on the toolbar.

For the Fernesck Mine, go through a similar process. Select suitable ambient and music tracks if the defaults are not appropriate. As an interior area, however, the Fernesck Mine doesn't really have a day / night cycle so it is usually best to play the same sound for both day and night.

Sound Objects: Fernesck Mine

Just as tile lights help draw the player's attention and serve to highlight particular regions of the area, so too, do sound objects. Placing sounds that imply action or movement that can't be seen can really bring an area to life. This can be done through single-shot and positioned sound-loops.

1. Switch to the Fernesck Mine area.
2. Open the Sound Object palette. (Third from the right on the bottom row.)
3. Select "Entrance Scary" from the Magical category and paint it close to the entrance to the mine. Notice the two concentric spheres. Zoom in a little to take a closer look. The model is a musical note on top of an arrow.
4. There is a lot going on here. This sound object is positional - it will sound as if it is coming from a specific point. The outer sphere indicates the maximum distance at which the sound can be heard, while anywhere inside the inner sphere, the volume is at maximum. The volume increases steadily as the player moves from the outer sphere to the inner sphere.

Tip: It may be necessary to turn off the ambient sound and ambient music while working with sound objects.

5. Select "Water Drips" from the Nature category. Paint it in the center of the tracks, close to the entrance to the mineshaft room. This is a random positional sound - it will sound as if it is coming from a specific point, but that point changes every time the sound is played.
6. The box indicates the range of where the sound will appear to come from, and the spheres mean the same as they do for standard positional sounds.
7. Select "Properties" from this sounds' context menu. Switch to the "Positioning" tab. Set the "Random Range" to 70m North-South and 5m East-West. This should be enough to cover the entire tracks. This means the sound should appear as if it is coming from anywhere inside the box.
8. Switch to the "Advanced" tab. Set the Interval to 8.00 seconds, and the Interval Variation to 2 seconds. All together, this means that the sound should come anywhere in the range every 6 to 10 seconds.

Note: Because the volume setting is not as big as the range, it is possible that the players may not hear the sound every time.

9. Select "Miner Pickings" from the Civilization category. Paint it anywhere in the area (close to the entrance is usually a good place). This is an area-wide sound, meaning it won't sound like its coming from a specific point.
10. Select "Properties" from this sound object's context menu. On the "Basic" tab, set the volume to about 7.
11. Switch to the "Advanced" tab, and set the Interval to 5 and Interval Variation to 2.

Tip: The key to using sound is to reinforce the plot and atmosphere for the player, to create illusions of actions that are not actually occurring in the game and to draw attention to people, places or objects with audio cues.

12. Select "Rope and Pulley Squeaks" from the Civilization category. Paint this instance close to the entrance. Notice that this is already configured to be a random positional sound, but in this case, it should be area-wide.
13. Select "Properties" from this sound object's context menu. On the "Basic" tab, set the volume to about 7.
14. Switch to the "Positioning" tab, and select "Plays Everywhere in Area".
15. Switch to the "Advanced" tab. Change the Interval to 10 and the Interval Variation to 3.

Tip: Balance the sounds in the area by adjusting distance and volume so that sounds come across as they would naturally. This means not having a distant birdcall at maximum volume while having a waterfall set close to zero. If the sounds in the area are balanced properly, it is easier for the player to pick out which ones are important and which ones fill in the background.

Extra Credit

Add some other sounds you might expect to hear in a mine, such as bats.

Note: A single-shot sound is a sound that plays once and has a definite start and finish.

Note: With this in mind, it's also possible to use too much sound in an area and clutter it up. When it comes to audio, sometimes less is more. Try varying the times between sounds playing and the pitch to give an area a more natural feel.

Add appropriate sounds to the Inn area, if it was created.

Tutorial 9: Scripts

Introduction

The NWScript language is very powerful, and you can do some truly amazing things with it.

The game attempts to run a script on an object instance whenever a particular event happens to it. Instances of every game object type have a set of events to which scripts can be assigned. In addition, each area and the entire module as a whole have certain events. The game does not require that every event have a script assigned to it.

The purpose of this tutorial is to introduce the NWScript language and the Toolset's Script Editor.

Goal

Attach a custom script to the module's OnAcquireItem event that updates the player's journal when the player picks up Jacen's ring.

Tutorial

OnAcquireItem Script

1. Start the Toolset and load the "tutorial" module.
2. Select "Module Properties" from the Edit menu to open the Module Properties window.
3. Press "Edit" for the OnAcquireItem event on the "Events" tab to open the Script Editor with a new script.
4. Enter the following text into the script, completely replacing whatever contents were there originally.

```
void main()  
{  
    // variable declarations  
    object oPC; // the PC that acquires the item  
    object oItem; // the item that was acquired  
  
    // get the item that was acquired  
    oItem = GetModuleItemAcquired();
```

```

// if this item is a valid object
if (GetIsObjectValid(oItem) == TRUE)
{
    // if the valid item's tag is "it_RingJacen"
    if (GetTag(oItem) == "it_RingJacen")
    {
        // get the PC that acquired the item
        oPC = GetItemPossessor(oItem);

        // on the PC, set the journal with the tag "jt_Falstadd"
        // to its second entry
        AddJournalQuestEntry("jt_Falstadd", 2, oPC);

    } // if
} // if
}

```

5. Save the script as "me_acquireitem" and close the Script Editor.

Note: A warning should appear saying that scripts must be compiled before they will work in the game. Unless this option is disabled (not recommended) the script editor will automatically compile every modified script.

6. Press "OK" to close the Module Properties window.

A Custom "Action Taken" Script

1. Open Falstadd's conversation by selecting "Edit" from its context menu in the module contents list.
2. Select Node 3 ("Thank you for..."). Switch to the "Action Taken" tab, and press the Script Wizard button.
3. Check the "Give rewards" checkbox and press "Next".
4. Enter 2000 (based on the value entered in the Journal Editor for this quest) in the "Give XP" field, check the "Give To Party" checkbox beneath it and press "Next".
5. Name the script "at_falstadd03", check the "Start the Script Editor" checkbox and press "Finish".
6. Once the script editor opens, the script should appear like this:

```

void main()
{
    // Give the speaker some XP
    GiveXPToCreature(GetPCSpeaker(), 2000);
}

```

7. The only problem with this is that if the reward amount specified for this quest in the Journal Editor changes, this script must be changed and recompiled, too. A better way would be to determine the amount of experience currently assigned to

the journal category while the module is being played and use it instead. This requires making the script look like this:

```
void main()
{
    // variable declarations
    int iXP;

    // determine how much experience this quest awards
    iXP = GetJournalQuestExperience("jt_Falstadd");

    // give the speaker some XP
    GiveXPToCreature(GetPCSpeaker(), iXP);
}
```

8. Once the script has been modified, save the script, close the Script Editor, save the conversation and then close the Conversation Editor.

Save the module, close the Toolset and try running through the module from beginning to end.

Extra Credit

Create a conversation for the miners. The miner to whom the PC is speaking should be grateful for being rescued and the PC should tell them to leave the mine immediately. The PC should be awarded 50 XP and the miner should then immediately head towards the exit and back to Fern. If the PC talks to the miner again, the miner should respond with more thanks but take no further action.

It should be possible to do this by creating three scripts, two with the Script Wizard (requiring only slight modifications) and one custom script. First create the conversation. It will be similar to Falstadd's conversation in that the first time the PCs speak with him he will say one thing and on subsequent attempts he should say something else.

The two scripts that control what the miner says should set and check a local variable on the miner. These scripts should be created using the Script Wizard, like those in Tutorial 6 and modified so that instead of using the GetPCSpeaker() function, they use the OBJECT_SELF constant. OBJECT_SELF simply defaults to the object that is running the script, which in this case is a miner. This allows the PC to rescue multiple miners and receive a reward for each one. If the variable were on the PC, only a single miner could be rescued because the check on subsequent miners would indicate that the PC had already spoken to him.

The third script should be run on an action taken from a conversation node that is only accessible the first time the PC talks with the miner. This script should award the PC 50 XP and then have the miner to move towards the exit and into town. The easiest way to move a creature is to have it move to another location or object. Now, use the

GetObjectByTag, AssignCommand and ActionMoveToObject functions to have the miner run to Falstadd.

Don't forget to use OBJECT_SELF as required and to assign the conversation to the miner blueprint!

What Next?

Check out <http://nwn.bioware.com/builders> for more information on creating adventures.