The Titan Quest Editors

A Modder's Guide

Version 2.0 Last Updated 8/9/2017

Table of Contents

Contents

Intro: Meet Your Tools

A set of modding tools are installed by default with your copy of Titan Quest. These tools are the same ones used to create Titan Quest and as a result, they're quite powerful!

- Art Manager - Homebase for managing assets in your mod
- Editor – Create levels
- Quest Editor – Create quest logic

Some of the tutorials in this guide use tutorial projects, which are installed in the mod project working folders as described below. They are numbered the same as the tutorials here; so Tutorial01 will match the first section below, and so on. If you're feeling adventurous, feel free to browse the projects, test them for yourself, and try to understand how they work.

Setting Up the Tools

These tools can be found via your start menu, under 'Programs -> THQ -> Titan Quest -> Toolset'. To get started, go there now and run the Art Manager. You will prompted to choose your working directories. The default values are good – just click 'OK'.

The Art Manager will open and the working directories will be set up under 'My Documents -> My Games -> Titan Quest > CustomMaps'.

Next, go to the Help menu and select 'Install Tutorials'. This will copy the tutorial files that you should use while reading this guide. Everything should now be properly set up. Proceed when ready.

Tutorial 1 : Basic Navigation and Object placement

This first tutorial is an introduction to making your way around the editor and setting your level up so it can be playable in-game. We're not going to do anything fancy just yet, we'll just be setting up a skirmish against a couple of monsters so we can get into the game and fighting as soon as possible.

Loading a World in the Editor

The first thing you're prompted to do when you fire up the Titan Quest Editor is select a world to open.

 1) Under 'Mod Name', select 'Tutorial01' from the dropdown. 2) A directory tree will appear. Expand "Maps" and select "Tutorial-Basic_Navigation.wrl" 3) Click 'OK'
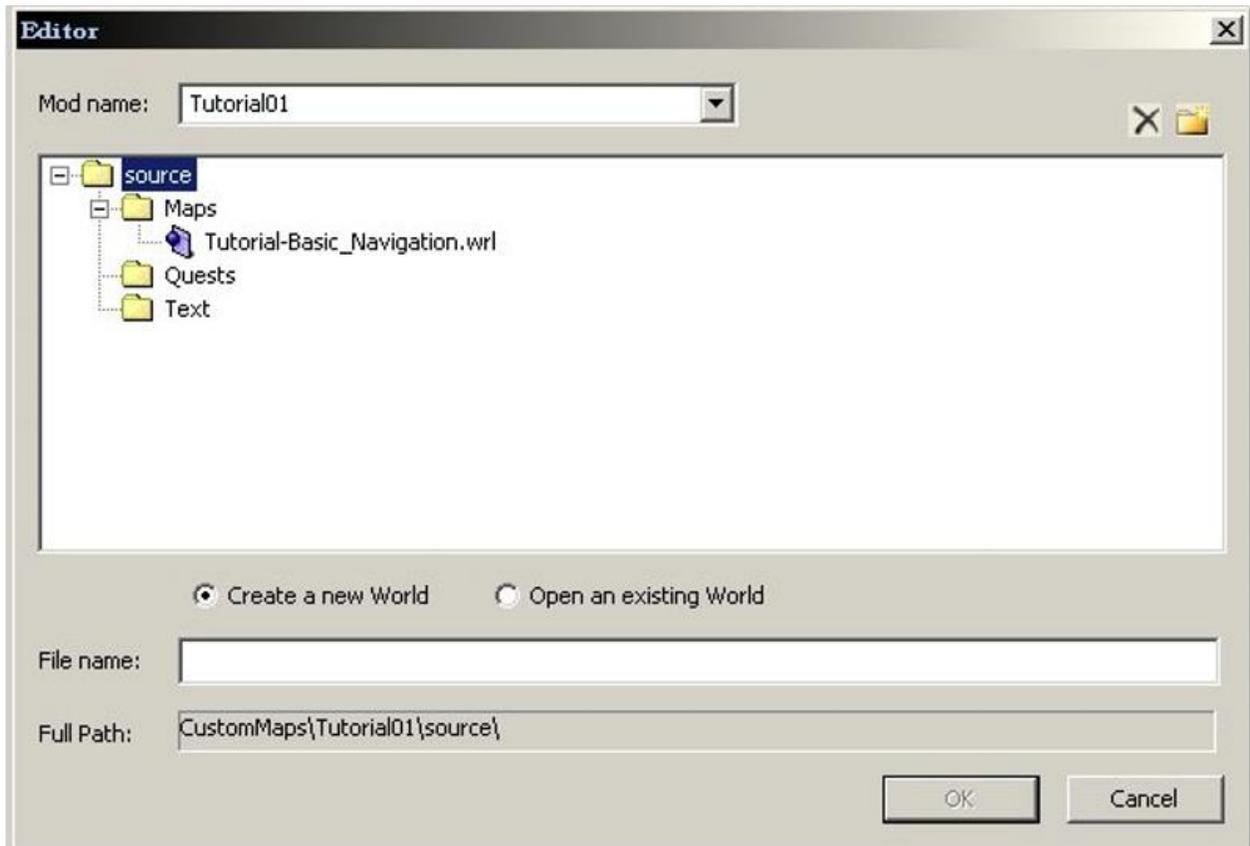
1.1 Navigating the Editor

1.1.2. Navigating in Layout Mode

When the world loads, the first screen you will come to is the Layout Mode screen. This will show you the world and all of the levels within it. A level is a small segment of the world that can be moved around like a puzzle piece. By splitting the world up into levels, the game is able to stream content as the player navigates a world without pausing to load for long periods of time.
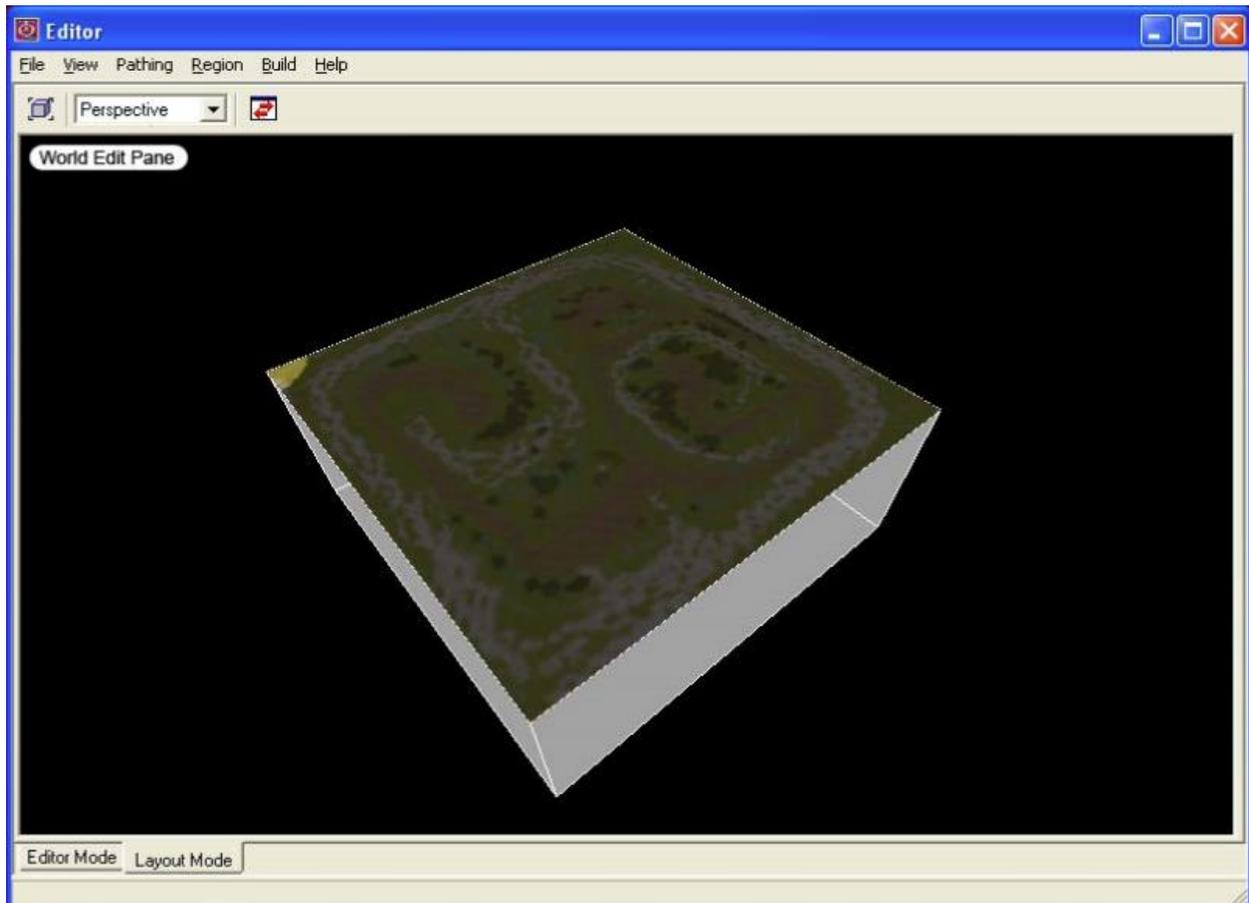
As your worlds get bigger, you will need to move the camera around in Layout Mode to get a better view of the arrangement of levels across the world.

1) To pan the camera across the world, hold the right mouse button and move the mouse.
2) To rotate the camera around the world, hold down the middle mouse button and move the mouse left or right.
3) To zoom the camera in and out in Layout mode, roll the mouse wheel up and down.

1.1.3 Choosing a Level to Edit

1) Select the box in the middle of the layout mode screen. Notice that it becomes highlighted and its name is superimposed over it. Layout mode is good for placement of levels and getting a feel for where different regions will go in relation to everything else. Now that we want to get to actual level editing, we need to enter Editor Mode. We can do this in either Layout Mode or Editor Mode.

Layout Mode:

1) Select a level to edit.
2) Press the mode change button.

Editor Mode:

1) Select the Editor Mode tab at the base of the screen.
2) Once you are in Editor Mode, use the rolldown list in the top-right corner of the window to select the level you wish to edit.

1.1.4 Navigating in Editor Mode

After a brief loading period, you should be looking at some grassy hills like those pictured below. This is the level we will be editing. As the whole level does not fit onto the screen
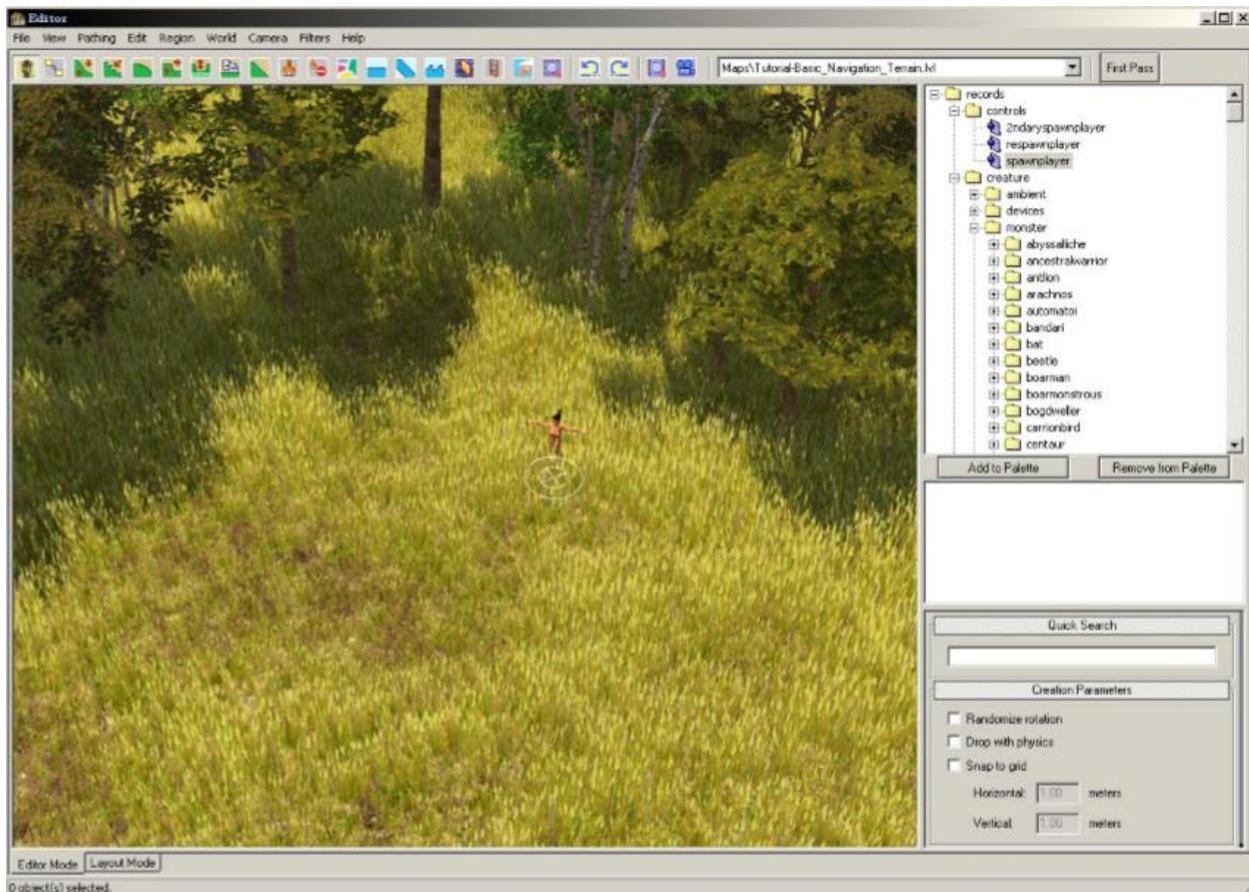
1) To pan across the surface of the terrain, hold the right mouse button down and move the mouse. 2) To rotate the camera around a spot on the terrain, hold the middle mouse button down and move the mouse left or right. 3) If you need to zoom in and out, you can roll the mouse wheel up and down to get closer or farther away from the ground.

1.2 Object Placement

Example 1.2.1 Object Placement Now that we know how to find our way around the level, let's start placing some objects. To place an object, you must select what type of object you want and then click where you want it to appear on the terrain in the Level Edit Pane. We'll start off by placing a few satyrs.

1) Select the Object Placement Tool .
2) In the directory pane on the right, locate and select the object 'records -> creature -> monster -> satyr -> am_peltast02'.
3) Click on the terrain in the Level Edit Pane to place the satyr.
4) To place another satyr, click again on the name of the object in the Directory Pane.
5) Then, click on the terrain in the Level Edit Pane.

Tips

- Multiple copies of the same object can be placed by holding control while left-clicking.
- Checking the randomize rotation box in the right pane will drop objects with a randomized orientation. This way multiple objects won't be aligned in exactly the same way.
- Checking the snap to grid option snaps the placement of the object to the nearest intersection points on the height map grid. This is especially useful for lining up things like walls and buildings.

1.2.2 Object Deletion Deleting objects is pretty intuitive.

1) Select an object.
2) Press the Delete key on the keyboard.

1.3 Object Movement

So what happens if we place an object, and we want to move it somewhere else? There are three types of movement for an object: lateral movement, vertical movement, and rotation.

1.3.1 Lateral Movement of Objects determines where on the terrain the object will be.

1) Left click on an object to select it. A movement indicator will appear at the base of the object.
2) Move the mouse to the center of the symbol (where the cross is). You'll notice this causes the mouse icon to turn into the four-way arrow icon.
3) When this icon appears, you can click and drag the object across the terrain.

1.3.2 Vertical Movement of Objects Vertical Movement determines how high above or below the terrain the object will be.

1. Select an object.
2. Move the mouse so that it is resting on the inner circle of the indicator. You will notice that the mouse icon changes to just an up-and-down arrow.
3. At this point, by clicking and dragging the mouse up or down, you can move the object up into the air or down below the ground.

1.3.3 Rotation of Objects

Rotation determines which direction the object will be facing.

1. First, select the object.
2. Move the mouse to the outer circle of the movement indicator. The mouse pointer should change to an arrow chasing it's tail.
3. Click and drag the mouse left and right to change the direction the object is facing. Bear in mind that this mode isn't very sensitive to mouse movement, so you'll have to drag a long way to do a 180.

1.3.4 Aligning Objects to Ground

Pressing the Spacebar will move all selected objects so that they are the same level as the ground under them. Note that this will only work for the terrain level. If you want to place objects on other objects, you'll have to eyeball it.

1.3.5 Selecting Multiple Objects at Once

There are two ways to select multiple objects in the editor: the selection box, and Ctrl + left-clicking.

1. Selection Box
2. While left-clicking, drag a large selection box by moving the mouse.
3. All objects within this box will be selected.

 Ctrl-Clicking

1. Left-click an object.
2. While holding the Ctrl key, left-click another object.
3. Each new object is added to the group.

1.3.6 Copy and Pasting Objects

Objects in the game can be copied and pasted to other locations in the world.

1. .Select an object or objects.
2. While selected, press Ctrl + C
3. Press Ctrl + V to paste the object(s) in the world. Pasted objects will appear in the center of the screen.

 You can highlight multiple objects by drawing a box, with the mouse or clicking on an object and then clicking on other objects while holding Ctrl key. These objects can be moved as a group laterally, vertically, or rotated around a central axis.

1.4 Spawn Points

So now we have enemies, but where does the player appear? We specify where the player appears with a spawn point. The spawn point is where the player will enter the world when the game begins (the model used to represent the spawn point has no effect on what your character will look like). Spawn points should not be confused with respawn points, which are objects that, when activated, act as a place for the player character to reappear after they die. We will discuss respawn points more in Tutorial 7. If no respawn points have been activated, the player will also respawn here when they die. You always need to have at least one spawn point in a world or your character will never appear, and you will not be able to start your mod.

Placing a Spawn Point

1. In the Directory Pane, navigate to the folder 'records -> controls'.
2. Select the object labeled 'spawnplayer'. A 'spawnplayer' object is placed and manipulated like any other object.
3. Click on the terrain in the Level Edit Pane to place a spawn point (see section 1.3).

1.5 Completing Your World and Levels

So we've got all our objects placed, let's make our way towards going into the game and tearing it up! We have a few things we need to do first. In the Editor, we need to build pathing meshes and maps. Pathing meshes determine where and how players and other actors can move about the level. The 'Map' in this case, is the image displayed on the minimap in the in-game user interface. These are both done in Layout Mode. You will need to rebuild pathing meshes and maps any time you edit the terrain or move objects around the map.

1.5.1 Building a Pathing Mesh

1. Enter Layout Mode by clicking the 'Layout Mode' tab at the bottom of the window.
2. Select 'Build -> Rebuild All Pathing' from the Menu Bar. A progress bar will appear to give you an indicator of when it's done.
3. Press 'OK' to close the Level Progress window.

1.5.2 Building Maps

1. In Layout Mode, select 'Build -> Rebuild All Maps' from the Menu Bar. A progress bar will appear to give you an indicator of when it's done.

1.5.3 Save Any Changes 1) Select 'File -> Save All' in the menu bar. Once you've saved any changes to your world and all your levels, we're done in the Editor. But we've still a few steps left to go.

1.6 Building a Map

The next couple of steps will involve Titan Quest's Art Manager. Most often, we'll be using the Art Manager to take our raw source data (.wrl, .jpg) and bake it into a file asset that is optimal for Titan Quest (.tex, .map). Titan Quest reads in environment information from the .map file type. To make a map, a designer must take a .wrl and any .lvl's used in the world and bake them into a Map. In the interest of avoiding the more complicated steps this early, a .map has already been created that includes the .wrl and .lvl that we were using in the editor.

Example 1.6.1 Building a .map Start by finding the folder the .map is in.

2. Open up the Art Manager and select "Tutorial01" from the Mod menu.
3. Select Build menu and click on Build. You will see some activity in the pane at the bottom of the screen. When the Art Manager is done processing the .map, it will tell you the total build time for the action.

1.7 Playing Your Map

So, our map is all set! Let's go into Titan Quest and try it out.

1.7.1 Playing a Custom Map

1. Fire up the main Titan Quest game.
2. Select Play Custom Quest from the main menu.
3. Select "Tutorial01 ~ tutorial_basic-navigation.map" from the list 4) Click the Start button to play your map.

Remember

1. You must always place at least one a Spawn Point on your map
2. Build pathing and map when you make changes to either
3. After you have saved your map, you must re-build your mod to see the changes in-game

Tutorial 2: The Art Manager

The Art Manager is used to take raw data and builds it into a format that is readable by the main Titan Quest game. Think of the Editor, Quest Editor, etc. as mixing bowls where ingredients are combined and the Art Manager as the oven where these ingredients are baked into an edible cake. Data created in the Editors is called source data. The art manager turns this source data into an asset. The Art Manager is also used to create Database Records. Database records are ways of arranging assets into objects that the player can interact with in-game.

2.1 Navigating The Art Manager

1) Open up the Art Manager. The pane on the left is the Directory Pane. It is used for navigating through the different directories that hold the source files, asset files, and database files. The user can switch between source, asset and database files with the tabs underneath the Directory Pane. The contents of the folder selected in the Directory Pane are displayed in the Folder Pane on the right.
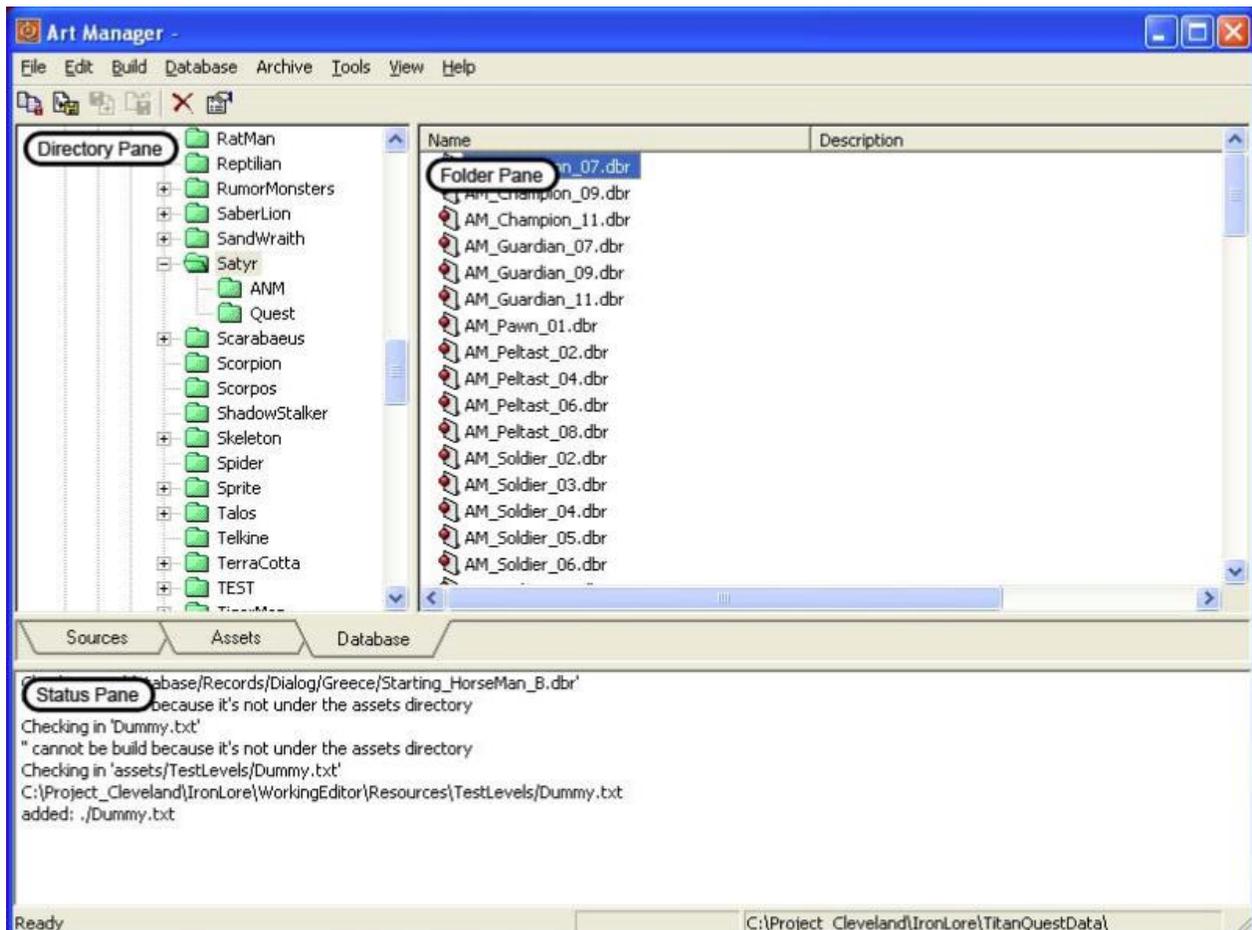

Below the Directory Pane and Folder Pane is the Status Pane. The Status Pane gives the user helpful information about the progress of any actions the Art Manager is taking.

2.2 Making Assets in the Art Manager

Source files placed in your mod's assigned folder or a sub-folder should automatically appear under the Source tab of the art manager.   When you select the mod in the Editor, it will automatically save your files to the correct directory.

2.2.1 Auto-creating an Asset

Usually, you can auto-create assets. In the Source tab, select a quest or world file and right click on it. Click on the "Auto-Create Asset" option on the menu that appears. That file will automatically have the appropriate asset (with the same name) created under the asset tab.
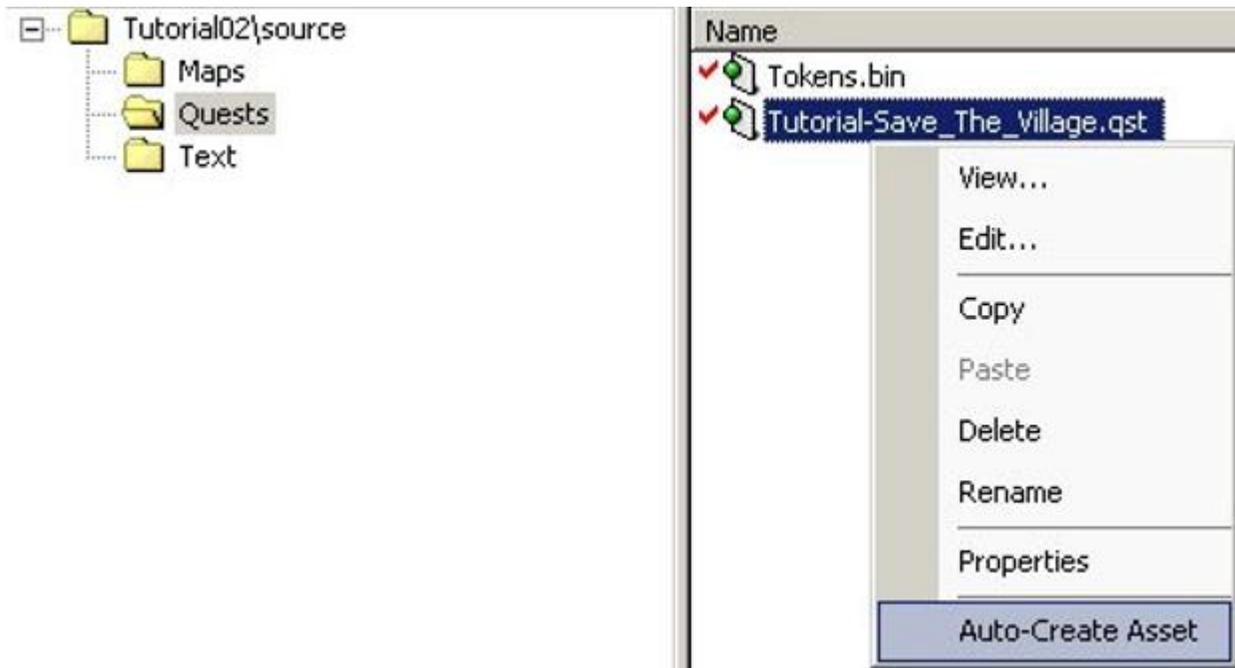
1. Open up the Art Manager and select "Tutorial02" from the Mod menu
2. Expand the 'Tutorial02\assets' folder 3) Select the "Quests" folder and right-click on "Tutorial-Save_The_Village.qst" and select "Auto-Create Asset."
3. You will be prompted for the asset name (it should choose the correct) one. Press "OK". The asset will be created in the correct folder in the assets tab. 4) That was the quest, now for the map. Right-click on the '.wrl' file and auto-create its asset. It and the accompanying level files will be automatically incorporated into a '.map' asset.

2.1.2 Manually Creating Assets

1. Assets can also be created manually:
2. Click the Assets Tab
3. Create any folders needed to mirror the location of the source file
4. Select the folder and right-click the asset pane.
5. Select "New" and choose the asset type to create. From here, you can select the source manually, and the asset will be created.
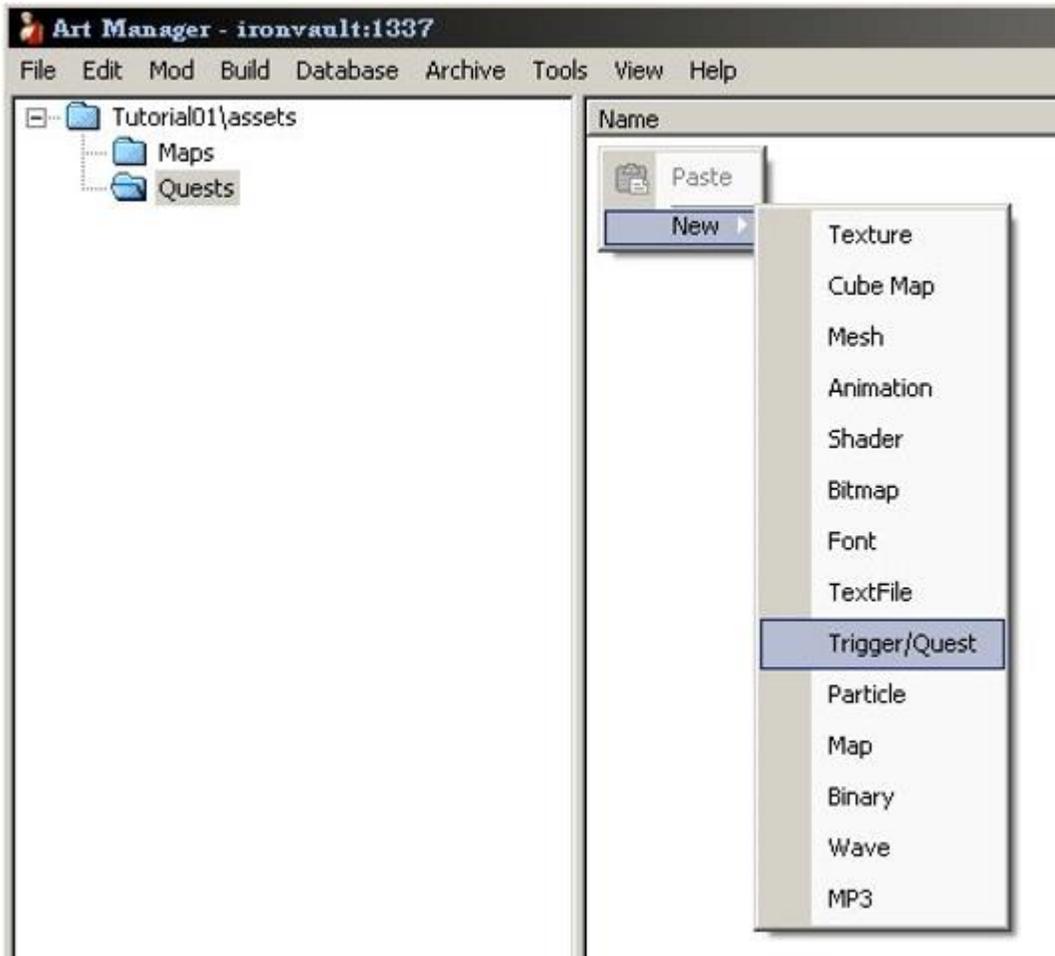
2.2 Building in the Art Manager

In order to see you changes in effect, you must build your assets. Building "cooks" the changed source files and database records into the format that can be read by Titan Quest.

In general, if you've made multiple changes, you will want to rebuild your entire mod – it won't take too long. Go to "Build -> Build". A word of warning: Don't close the Art Manager until it is finished building!

You can also build assets, database records, and directories under either tab individually. Right click on the item or folder and select build from the menu that appears.

Tutorial 3 : Basic Terrain

Alright, so the last tutorial taught us a few of the steps necessary for making a level that is playable in-game. We also learned about customizing levels by changing the arrangement of characters and scenery. This tutorial will tell us how to sculpt the terrain to make our own personal world.

3.0.1 Checking Out the Environment

1. Open the Editor.
2. Select mod "Tutorial03"
3. Open up 'Tutorial-Basic_Terrain_Terrain.lvl.wrl' in Editor Mode.

The Bump/Dip Tool

The first and most basic sculpting tool available is the Bump/Dip Tool. This is used to raise and lower the terrain under the paint brush.

3.1.1 Using the Bump/Dip Tool

1. Select the Bump/Dip Tool in the toolbar.
2. Move the mouse over the terrain in the Level Edit Pane. You will see two concentric circles with a small X in the center being projected onto the ground. This is the tool's paintbrush. It is a visual representation of where changes will occur when the tool is used.
3. Left-click and hold on the terrain. A mound rises below the paintbrush.
4. Left-click and slowly drag across the terrain. A dune starts to rise and extends in the path that the paintbrush travels.
5. Hold the Shift key, click and drag. The ground below the paintbrush is lowered.
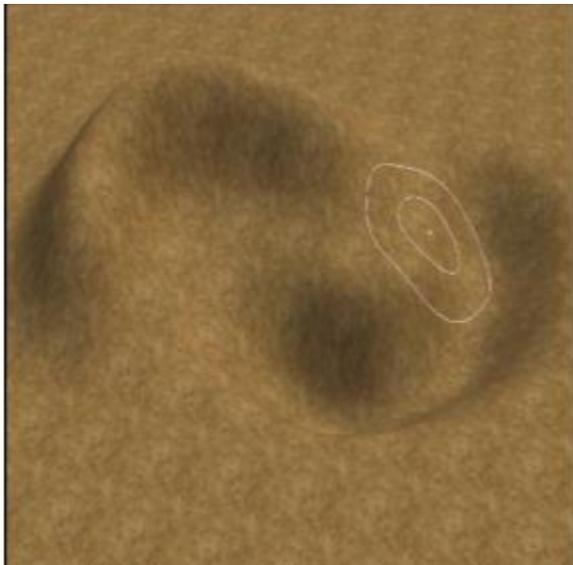
3.2 Paintbrush Properties

The effect of several tools can be changed by changing its brush properties. If a tool's paintbrush properties can be changed they will appear in the Brush Properties Pane on the right side of the screen. All of the brush properties can be changed by typing in new values into each field. The three standard brush properties are Radius, Strength, and Speed.

Radius

The first property is Radius, which affects the size of the area the paintbrush will affect.

1. Enter a new value into the Radius field in the Brush Properties Pane. The size of the circles on the terrain will change. The paintbrush's radius can also be changed with hotkeys.
2. Press the '[' key. The paintbrush's radius will shrink. 3) Press the ']' key. The paintbrush's radius will grow.

Strength

The second paintbrush property is Strength. Strength describes how much the brush will affect the terrain as it gets towards the edge of the brush. A low strength will have it pulling mostly the middle, and a high strength sculpts almost uniformly across the paintbrush.

1. Select the Bump/Dip Tool.
2. Enter a value of 10% in the Strength field in the Brush Properties Pane.
3. Left-Click and hold the paintbrush in one spot on the terrain. The terrain will rise very sharply in the middle of the paintbrush but the terrain near the edges of the paintbrush is almost unchanged.
4. Enter a value of 100% in the Strength field in the Brush Properties Pane.
5. Left-Click and hold the paintbrush in one spot on the terrain. The terrain under the paintbrush rises at a nearly uniform rate.

 Speed

The last property, Speed, affects how quickly the brush pushes and pulls on the terrain.

1. Select the Bump/Dip tool.
2. *Enter a value of 10 into the Speed field in the Brush Properties Pane.*
3. Left click and hold the paintbrush in one spot on the terrain. It takes several seconds before there is any noticeable effect on the terrain.
4. Enter a value of 500 into the Speed field in the Brush Properties Pane.
5. Left click and hold the paintbrush in one spot on the terrain. A large hill quickly rises out of the terrain.

![Noise Tool icon]	The Noise Tool

The next tool is the Noise Tool. The Noise tool creates a series of bumps and dips under the area of the paintbrush. This can help add some flavor to unnaturally smooth terrain. It features the three standard brush properties.
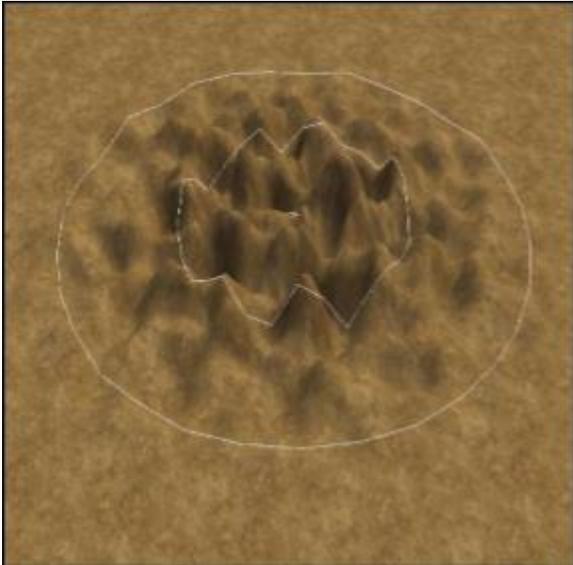
![Smooth Tool icon]	The Smooth Tool

After that, we have the Smooth Tool. Smoothing helps average out the slope of the terrain under the paintbrush. This is useful for making lumpy terrain more (wait for it...) smooth. It has the standard brush properties of Radius, Strength, and Speed.
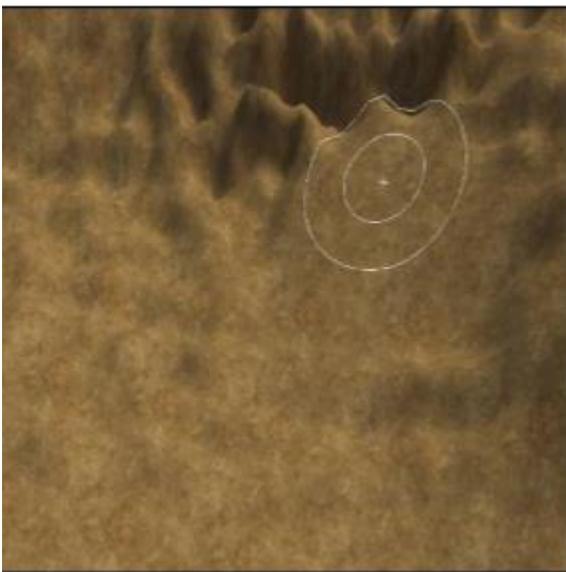
	The Plateau Tool

Next up is the Plateau Tool. The Plateau Tool moves all of the terrain under the paintbrush into a flat area at a specific height.   In addition to Radius, Strength, and Speed, the plateau tool also has the Height Property. The Height property determines the height at which the flattened terrain will be raised to. One way this can be changed is by typing a value into the field.

1. Select the Plateau Tool.
2. Enter the value "10" into the Height property in the Plateau Properties Pane.
3. Left-Click and drag the paintbrush across the terrain. A plateau rises up along the path of the paintbrush. A the plateau's height can also be determined by sampling the height at a specific point on the terrain.
4. Move the paintbrush so that the center of it is about halfway up the side of the plateau.
5. Hold the Shift key. 6) Left-click. The height property will change to the height of the point you shift-clicked on. The Height property can also be changed visually.
6. Hold the Control key.
7. Move the mouse up and down. A wireframe cylinder will rise up or drop into the ground to give you a visual representation of how high the plateau will be.
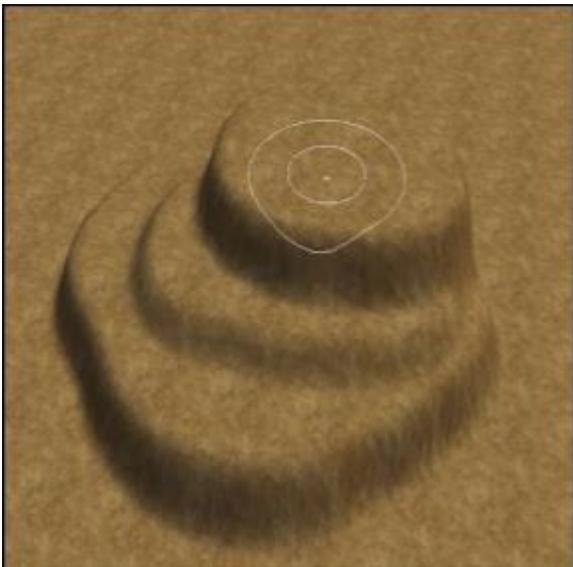
	The Relative Plateau Tool

 The next tool is the Relative Plateau Tool. This raises all of the terrain underneath the paintbrush up a specific amount. This is helpful if, for example, you sculpted some nice terrain, but decide you needed it all to be 10 meters higher. Rather than use the Bump/Dip Tool to try to raise everything 10 meters and risk losing the shape of the terrain, you can just use the relative plateau tool and paint the area to be 10 meters higher. In addition to the standard brush properties, Relative Plateau has another property called Offset. This determines how high up the relative plateau will be in comparison to where it was before.
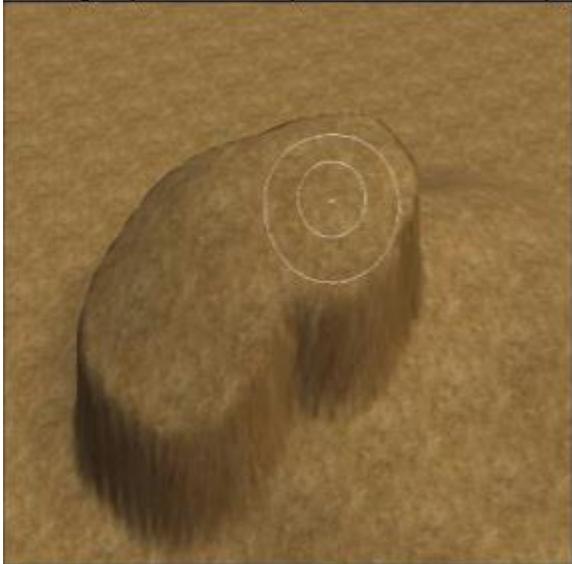
The Terrain Copy/Paste Tool

The Terrain Copy/Paste Tool is good when you want to have two similar areas in your world, but don't want to have to sculpt the terrain twice.

1. Select the Terrain Copy/Paste Tool. When you have the tool selected, you'll notice the paintbrush has an additional square around it. As with other brush-based tools, the paintbrush for the Copy/Paste Tool can be made wider or narrower by pressing the '[' or ']' key respectively.
2. Move the paintbrush so that the square contains all of the terrain you want to copy.
3. Left-click the mouse. Notice the paintbrush has turned into a ghost image of the shape of the terrain you copied. If need be, you can also rotate the ghost image of the terrain.
4. Press ',' or '.' to rotate the terrain counter-clockwise or clockwise respectively.
5. Move the paintbrush to the area where you want to paste the the copied terrain.
6. Left-click the mouse to paste the copied terrain.
7. To clear the ghost terrain image and terrain clipboard, press Spacebar.
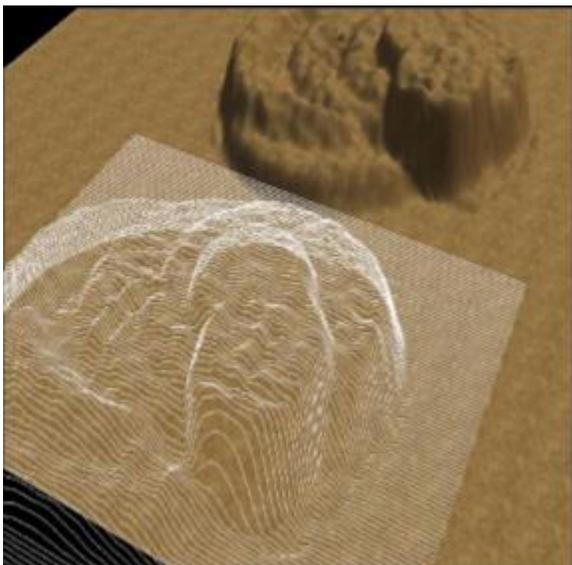
The Ramp Tool

The Ramp Tool is a way to create a smooth incline from one elevation point to another.

To create a ramp, just click and drag from one point on the terrain to another. The editor will automatically create a smooth ramp between those two points of different elevation. (Which means creating a ramp on flat ground will have no effect!) The width of the ramp can be set in the Ramp Properties Pane on the right.

3.9 The Texture Paint Tool

In addition to sculpting the terrain, we can also paint textures onto the terrain to give it more character.

Adding a Texture Layer

1. Start by selecting the Texture Paint Tool. Each terrain texture is a different texture layer. The user paints the opacity for various areas of each layer. Areas of a texture with a low opacity will show more of the layers below than an area with high opacity. The texture layers the user is currently using are displayed in the Texture Layers Pane on the right side of the screen. The layers that are lower on the list will be drawn on top. So if the list had, in order from top to bottom, rock, dirt and grass in it, it would draw the rock, then the dirt on the rock, and then the grass on top of the dirt.
2. To add a layer, click the 'Add Layer' button underneath the list box. This opens up the Texture Picker window.
3. Navigate through the folders in the Directory Pane on the Left. Each of the textures present in that folder are listed in the Folder Pane on the right with a small thumbnail of the texture.
4. Select a texture in the Folder Pane. A larger sample of that texture appears at the base of the Texture Picker window. 5) When you are satisfied with your texture choice, click 'OK'.

Texture Picker

Directory Pane

- dirt
- egypt
- fieldarid
- forestconifer
- greece
  - beach
  - city
  - farm
  - forestcon
  - forestdec
  - frozen
  - grassland
  - highland
  - knossos
  - olympus
  - paved
  - river
  - rock
  - swamp
  - wasteland
- joshtest
- orient
- tom
- tq2000

Folder Pane

...wers.dbr
changangrass.dbr
changanmud_rich01footprint.dbr
changanriver reeds.dbr
changanriver reedsthick.dbr
grassdebris01.dbr
grassdebris01.dbr
grassdebrisroots01.dbr
grassgreentall01.dbr
grassgreentall02.dbr
grassgreentall02a.dbr
grassrivertall01.dbr
grassswamptall01.dbr
grounddebris01.dbr
grounddebris01.dbr

Texture Sample

OK   Cancel

3.9.2 Painting With a Texture Layer

Terrain textures are painted directly onto the terrain.

1. Select the Texture Paint Tool
2. Select a texture layer from the Texture Layers Pane.
3. Move the paintbrush over the terrain
4. Click and drag to paint the area you want the texture to appear on.

In order to blend the textures together, the user can adjust the opacity of the paintbrush in the Texture Opacity Pane on the right. Each texture is designed such that as the opacity goes down certain areas disappear first, so pebbles will stay visible as dirt disappears, and foliage will shrink. Opacity can be changed by entering a value into the Opacity field in the Texture Opacity Pane dragging the slider next to the field. The opacity can also be set to the multiples of 10% by using the number row on the keyboard.

Tutorial 4 : Quest Objects

So we know how to create a world, make interesting terrain, and throw some enemy dudes at the player. But what if the player wants more motivation for killing monsters than money and XP? For that, we need to assemble quests. Each quest file helps organize both the actions that the game needs to take to progress and the conditions that trigger these actions. Triggers can be set up to do everything from opening doors, to killing objects, to giving Skill Points. For this example, we are going to be working with a simple "go kill the monster that's terrorizing the people" quest. The player starts off by talking to a Villager. When the player talks to the Villager, the Villager tells the player to go kill the Monster. The player then goes and kills the Monster. When the player returns to the Villager, the Villager gives the player a reward. The quest and all related database records have already exist. All we need to do is place the objects used in that quest into our map, and the quest will run when a player plays on our map.

Setup

1. Open the Art Manager
2. Under 'Mod' go to 'Select' and choose Tutorial04
3. Go to the 'Build' menu and select 'Build'   Now the database records are accessible in the Editor, and we're ready to go.

 4.0.1 Checking Out the Environment

1. Open the Editor
2. Select the 'Tutorial04' mod, and open 'Tutorial-Quest_Objects.wrl.' We have a world with some nice terrain and a few decorations. More importantly though, we have a quest object that is vital to this quest. This quest object is the Villager. The Villager is the fat old man standing next to his hut. However, we're missing an object that is vital to this mission. If the Villager is going to tell the player to go kill the Monster, we had best make sure there is a Monster present for the player to kill.

4.0.2 Placing a Quest Object For this tutorial a unique object called 'Tutorial-Quest_Satyr' was created. He is just like an 'am_peltast_02' type satyr, but this particular Quest Satyr is necessary to complete this particular quest.

1. Load up the level 'custommaps\Tutorial04\source\Tutorial-Quest_Objects_Terrain.wrl'.
2. Select the Object Placement Tool.
3. Expand the "Monsters" tree, and select "tutorial-quest_satyr" (All the standard Titan Quest assets are under the "records" tree. When you create your own database records, they will appear under the folders you create them in, and appear around the "records" folder. When you import records, you should move it to your own folders, otherwise it will appear hidden somewhere deep in a subfolder of a subfolder under "records".)
4. Place the 'Tutorial-Quest_Satyr' anywhere in the level you like simply by left-clicking at that spot on the map. You can alter the player's experience just based on where you put it. The player may have to battle their way up the side of a mountain or face an ambush in the valley

depending on which you think would be more fun. The point is, the trigger for this particular quest is based on a specific object type.

4.0.3 Build and Save the Map

Don't forget to build and save when you're done editing the world.

1. Rebuild all pathing meshes and maps if you changed the scenery or terrain at all.
2. Select 'File -> Save All' in the Menu Bar in the Editor.
3. Open the Art Manager.
4. 4) Build all.
5. 5) Try out your map in game.

Tutorial 5 : Advanced Terrain

So the last time we were learning about terrain, we learned the basics of sculpting terrain and adding some kick with terrain textures. Now we're going to get fancy! We'll learn about sector painting, a couple of water tools, and movement blocking zones.

5.0.1 Checking Out the Environment

1. Open the Editor.
2. Select mod Tutorial05
3. Find and open up 'Tutorial-Mountain_Stream_Terrain.lvl' in Editor Mode.   You'll see that the level is a large plateau with small bumps in the distance. We'll put some water in the pond and the river. Then, with the help of height fog we'll make it look like a misty river valley. You'll also see a trench running along the big plateau into a pit. This will become a river leading to a pond.

5.1 The Water Tool

Now that we're finished with the clouds, let's move onto the pond.

Making a New Water Type

1. Select the Water Tool. Just as we did with the terrain textures, we need to start off by creating a new type of water.
2. In the Water Types Pane on the right, click 'Add'. A new water type will be created and selected.
3. Double-click the "New Water Type" you just created. The Water Type Editor Window will pop up.
4. Type "Standard Water" into the Name field.
5. Click 'OK'.

Painting the Water

Much like a plateau, water is painted at a specific height. The water's height is also obtained in a very similar manner to that of plateaus.

1. Hold Shift and click about halfway up the rocky part of the riverbank. The water's height will be set to the height of the point you clicked. You can also set the water's height by holding the Control key and dragging the mouse up or down.
2. Click and drag to paint the water's surface. Paint the water's surface over the top of the pit at the end of the trench. Don't bother painting water into the river trench just yet, we'll be using a special river tool to do that.
3. To erase some of your water layer, select 'None' under the Water types and paint normally.

5.2 River Tool

5.2.1 Making a River

Now that the pond's there, let's go ahead and create the mountain stream leading to it.

1. Select the River Tool. The river tool is tricky because instead of actually painting the river, you place points that it follows. Each of these control points has bezier handles which control how wide a turn the river makes as it approaches the control points.
2. Select a water type. If you do not have a water type selected, you will not be able to place any points at all. You may need to create a new water type in order to use the river tool. You may want to create a new water type if you want it to flow in a different direction than other bodies of water.
3. Hold Shift to enter point creation mode. 3) While still holding Shift, place a control point by clicking. Start at the bottom of the river, by the pond, to ensure that the water will flow in the correct directions
4. To lengthen the curve handles as the point is placed, hold the left-mouse-button down drag the curve handle to a point that results in the curvature you were hoping for.
5. To create a new control point, release the left mouse button and shift-click again in the area you want the new control point to appear.
6. If you feel your placement of a control point is completely unforgivable, you can press backspace at any point to delete the last control point you placed.
7. Create a series of points going down the length of the trench so that the river flows between the banks. You may choose to do some terrain editing to make the edges of the river and the riverbanks connect properly.
8. To create a new river, release the Shift key and click anywhere on the terrain. You may now place points normally for a new river.

5.2.2 Editing a River

You may decide you need to move the control points and curve points after they've been placed.

1. Hold the mouse over the center of a control point. The mouse will turn into a 4-directional arrow.
2. Click and drag to move the control point around.
3. Move the mouse to the top or bottom of the selection circle. The mouse will become an up-and-down arrow.
4. Click and drag to increase or decrease the height of the river above the terrain.
5. Move the mouse to the left or right of the selection circle. The mouse will become a left-right pair of arrows.
6. Click and drag to make the river wider or narrower at that point.
7. After selecting a control point, move the mouse over one of it's bezier handles. The mouse cursor will become a mouse chasing its tail.

8. Click and drag to move the bezier handle around. The curvature of the river will change as the bezier handles are moved and lengthened.
9. You can also widen or make the river narrower with the '[' and ']' keys when editing a point. Often times you can reduce the number of control points you need just by clever placement of curve points.

 Note: The Bezier curves in rivers can have a big impact on performance. We recommend to limiting your rivers to three or fewer Bezier handles. For longer rivers, you can string together rivers with two handles each.

5.3 Sector Painting

So let's start with the Sector Paint Tool. A sector is an area that changes the world properties when the player is standing in it. We will be manipulating the fog and lighting properties of the world in the sector we create in this example. In this case, when the player is running around on the high plateau, we want the base of the peaks to be hidden in a fog.

5.3.1 Making a New Sector Layer

1. We start by selecting the sector paint tool. One thing to remember is that the fog effects don't appear where the sector is painted. The fog effects appear everywhere when the player character is within the sector. We want the clouds to appear in the background when the player is running around on the main plateau, so we only need to paint the sector onto the main plateau.
2. Click 'New' to create a new Fog/Lighting sector layer. A new sector style entitled "New Sector Style" should appear. Now is an excellent time to give the new sector style a more descriptive name.
3. Enter the name "Mountain Fog" into the 'Style Name' text box in the Sector Paint Options Pane.

5.3.2 Sectors at Work

Let's give ourselves a quick demo of how sectors work.

1. Select the 'Mountain Fog' layer style.
2. Using the mouse as a paintbrush on the terrain, draw a quick patch of that sector by left-clicking anywhere on the terrain. You'll notice there's a small black sphere at the center of the screen. This is the object that the Editor uses to determine what sector your are in.
3. Move the camera so that the black sphere is just outside the terrain you painted. You will notice a Density slider in the Height Fog portion of the Fog/Lighting Options Pane.
4. Drag the Depth Fog's Density slider to a value of about .50.
5. Move the camera back so that the black sphere lies in the sector you painted. You'll notice that the screen fills up with semi-transparent fog.
6. Move the sphere outside of the sector. The fog will disappear. This same effect will happen in the game. When the character runs onto the sector, the fog effects will appear, and when the character leaves, they'll go away.

5.3.3 Painting a Fog Layer

Since we want the fog to appear in the background when the player is running around on top of the plateau, we should just designate the large plateau as the fog sector.

1. Select the sector layer you created.
2. Paint the fog sector such that it covers the entire plateau. Now, whenever the player is running around on top of the plateau, the fog effects for that sector will take effect. As nice as it is to have fog everywhere when we're standing on the mountain, it would look nicer if we could manipulate it so that each of the plateaus is poking out of the clouds. For that, we need height fog. Height fog creates a fog that gradually transitions from a specified density at one height to completely transparent at another height. Locate the Fog/Lighting Options pane on the right.
3. Slide the Depth Fog Density down to 0.
4. Locate the Density slider in the Height Fog portion of the Fog/Lighting Options Pane.
5. Set the height fog's density to .99.
6. Set the height fog's Bottom Height property to 1.
7. Set the Top Height property to 10. The fog will fade from .99 percent opaque at a height of one to completely transparent at a height of 10. It will appear as though the plateaus in the distance disappear into the clouds as the player runs through the sector.

5.4 The Impassible Area Tool

The last tool we are going to talk about in this tutorial is the Impassible Area Tool. The map-maker uses this tool to designate areas they don't want the player to be able walk around in.

Painting Impassible Area

This brush is used the same way the terrain painter tool is used.

1. Click and drag the paintbrush in the Level Edit Pane to paint the areas you don't want the player to be able to walk in.
2. The Impassible Area brush can be made wider or narrower with the '[' and ']' keys.
3. We don't want the player to be able to run down the sheer cliffs on the side of the map, so paint all of the rocky ridges around the edges of the plateau with the Impassible Area brush.
4. If you paint over some area you didn't mean to, hold down Shift and click to make your brush an 'eraser' that will remove the impassible area wherever you paint

So that's it. Don't forget to re-build your pathing, as the no entry brush affects that (see section 1.5.1). Just save in the editor and build in the Art Manager the way you would with any other map.

Tutorial 6: Dungeons and Grids

Up to this point, all of the areas we've been exploring have been outdoor environments. They take place in terrain type regions. Titan Quest also contains another type of region, known as a grid. A grid is essentially a board of tiles onto which the floor and walls can be painted. Grids are optimal for indoor, dungeon-type levels. Grids have three major components: floor, walls, and features. The ground is any area that the player is allowed to walk on. Walls can be drawn throughout the level to create a maze for the player, and features are other aspects of the dungeon that can merge with the walls and the floor. Features often contain portals which allow a grid to be connected to another region. We'll be talking more about portals in a later tutorial.

6.0.1 Checking Out the Environment

1. Open the Editor.
2. Select mod Tutorial06
3. Find and open up 'Tutorial-Dungeon_Crawl_Grid.lvl' in the Editor Mode. You'll see that there is already a little bit of dungeon floor stretching out across the screen. There are also some walls dividing the area into a few rooms.

 6.1 Grid Systems 6.1.1 Selecting a System 1) Select the grid edit tool.

 A purple plane with a lattice drawn onto it will appear. This shows the extents of the level grid. Floor and wall tiles cannot be painted beyond the bounds of the grid. The first step when designing a grid is to choose a system for your grid. A system is like an aesthetic theme for your grid. Each system has its own types of walls, floors and features that can be swapped for different art assets automatically by changing the grid's system. Let's try changing this dungeon's system. 2) Click the ellipsis next to the System field in the Grid Types Pane. This brings up a browser window of all of the available grid systems. 3) Browse for and select 'Records -> Underground -> Egypt -> EgyptCrypt.dbr'.

The dungeon will change from a series of natural, green walls to some brown, stone walls.

6.2 Painting a Grid

6.2.1 Painting the Floor We'll start off by painting some more floor area for the player to run around in. In the Grid Types pane on the right, you will see a pick box named Floor. 1) Select the 'Floor' Floor type. Like water, there is no erase option per se. You need to select the 'No Floor' Floor type in order to get rid of any floor tiles you wish to remove. 2) Click and drag the paintbrush over the tiles of the grid to paint floor tiles. Each tile you drag the mouse across should become occupied with a stone floor tile.

6.2.2 Painting the Walls

Walls are painted very similarly to floors.

1) Select the Wall type you want from the Walls pick box in the Grid Types Pane. Like the floor, there is no delete wall option, so you'll have to use the "No Wall" trick.
2) Click on the edge of a wall to place a wall segment. By dragging the mouse across the grid you will be able to draw a stretch of wall under the path of the mouse.

Placing Features

Another aspect that can be painted onto a grid is features. Features are special aspects of a dungeon such as doors.

1. Select 'Dng_Egypt01_Base_EntranceG_Int01' from the Features picker in the Grid Types pane.
2. Click to drop it on one of the tiles between the two torches at the end of the grid. Now the dungeon has a proper door.

 6.2.4 A note on multiple floors.

While it is possible to create dungeons with multiple floors using the editors, this feature is not supported with the provided tilesets. We recommend doing what we did – link multiple dungeon grids with staircases.

6.4 Finishing A World With a Grid

6.4.1 Placing Objects on a Grid Objects can be placed on a grid region the same way they are on terrain.

1. Select the Object Placement Tool.
2. Select an object to place from the Directory Pane.
3. Click on the grid to place the object.

6.4.2 Building a World With a Grid Worlds with grid regions are built the same way as any other world.

1) Build all pathing and maps.
2) Save all files.
3) Assemble the .wrl and .lvl's into a .map. Remember that if you split one of your grids into separate floors, each floor .lvl must be included in the .map as well.
4) Build the .map.

Tutorial 7: Object Fanciness Alrighty.

So we know how to place goodguys and badguys. We know how to move them and turn them around. But let's face it, there's more to a game than bad guys. These tutorials will delve more into placeable objects and their organization.

7.0.1 Checking Out the Environment

1. Fire up the Titan Quest Editor.
2. Select the mod Tutorial07
3. Find and open the level 'Tutorial-Fancy_World.lvl' in Editor Mode. You'll notice a waterfall falling into a pond. You'll also see some rolling hills with some dirt paths snaking through the terrain. Scattered along these paths you'll notice some clusters of circles. These are called Wander Points. Wander points are objects that NPCs wander between when they are idle. You'll also notice some NPCs who will ultimately be wandering between these wander points.

7.1 The Object Palette

Eventually you may find that there are one or two objects that you use often, but they are so far apart in the object tree that it is a pain to go and find them every time you switch between them. Fortunately, there is the object palette. The object palette is a place where you can place the objects you use most frequently for quick and easy access.

 7.1.1 Adding an Object to the Object Palette

In this example we will be adding a low-level satyr to the object palette.

1. Select the Object Placement Tool.
2. In the Directory Pane, find and select 'records -> creature -> monster -> satyr -> am_peltast_02'
3. Click the 'Add to Palette' button underneath the Directory Pane. 'am_peltast_02' will appear in the Object Palette.

 7.1.2 Placing an Object from the Object Palette into the Level Edit Pane The Object Palette works exactly like the Directory Pane. 1) Select the object in the Object Palette. 2) Click on the terrain in the Level Edit Pane to place the object.

7.2 Groups

 Groups are a way of grouping similar objects together. Often times an object needs to be put into a group of the appropriate type in order to function properly. We're going to create a group of NPCs who will wander around the map. There are already some NPCs placed on the map, but until they are placed in a group to make them wander, they will just stand still.

7.2.1 Creating Groups

To set your NPCs strolling, we'll need two sets of groups.

1. Select the Group Tool. You will notice in the Group List Pane on the right that there is already a group called "Wander Points".
2. Click the 'New' button in the Group List Pane. 33
3. In the Group Properties pane, change the Name to "The Wanderers"
4. In the Group Properties pane, change the Type to 'Npc Wanderers'
5. Next, create a new group. This one is for the wander points. Name it something like "Wander Here". The first group will contain the NPCs who are supposed to wander. The second group will link to the areas the NPCs are supposed to wander in.

7.2.2 Linking Groups

1. With the "Wander Here" group still selected, left click on each of the two wireframe spheres (called "bounding volumes"). Red boxes will appear around each of them, showing that they are now part of the group.
2. Select "The Wanderers" again, and left click on each of the three NPCs. Red boxes will appear around them when they are linked
3. Check the "Link Mode" box and left click on the black dot that appears between the bounding volumes. A large arrow will appear. This indicates that the NPCs are assigned to the wander points.

The grouped NPCs will mill about between the two wander points we've created. Generally, you should have at least two wander points, and you should place your wandering NPCs inside of them.

7.3 Sounds

To create a sound in Titan Quest, the level designer must place a sound object. As the player character approaches a sound object, the sound gets louder.

Creating a Sound Emitter

1. Select the Object Placement Tool.
2. In the Directory Pane, find and select 'records ->sounds -> soundobjects -> waterfall01'.
3. Place the object at the base of the waterfall. You will see a big green pillar. As the player gets closer to this object, the sound of the rushing water will get louder.

7.4 Respawn Points

A respawn point is a place where the player will reappear after they die. Respawn points cannot be used until the player runs into the object that represents the respawn point (a Rebirth Fountain, for example). When a player dies, they will reappear at the last respawn point they activated. For the designer,

creating a functional respawn point is a little more complicated than just selecting an object from the directory window and dropping it.

Creating Respawn Points

1. In the Directory Pane, find and select 'records -> item -> shrines -> respawntemplegreece01'.
2. Drop a few respawn temples on the terrain.
3. Select the Group Tool.
4. Create a new group.
5. Name the group "Respawn Shrines".
6. Designate the group to be of type 'RespawnShrine'.
7. Add the respawn objects to the group. Now, the respawn objects will be fully functional in-game. In general, it is a good idea to build your level first, then add and link respawn shrines after everything else is all set.

7.5 Treasure Chests

Implementing Treasure Chests

Treasure chests are objects that yield helpful items when the player interacts with them. When a player interacts with a chest, it plays an opening animation and several helpful items come shooting out.

1. Find and select 'records -> item -> containers -> greece -> monsterchest_epic_01a'.
2. Place the object on the map
3. Save and build the map.
4. Interact with treasure chest in-game. Note that treasures shoot out.

Tutorial 8: Traversing from Region to Region

So we know how to make terrain, and we know how to make grids. Getting from one terrain to another is easy, just put the two regions right next to each other. Connecting terrain to a grid or a grid to a grid is a little more complicated. For that, we need to use portals. Portals are essentially just doors from one region to another. Another way to move the player character around is to teleport to a specific point via a teleport trigger.

8.0.1 Setting Up 1) Open up the Editor 2) Select the mod project Tutorial08 3) Find and open the region 'Tutorial-Connecting_To_Dungeons.wrl' in Editor Mode. You'll see that it has a terrain region and a grid region. The terrain region has a cave sticking into a cliff face.

8.1 Portals

8.1.1 Creating a Portal Portals are built into certain types of objects. One of those objects is 'records -> underground -> naturalcave -> greecedngentrance_a01_ext.dbr', which is the cave object buried in the cliff face in the terrain region. That's one of the portals we'll be linking, but we still have to create one on the grid. 1) Select the Grid Edit tool. 2) Click on the grid region 'Tutorial-Connecting_To_Dungeons_Grid.lvl'. One way to create a portal on a grid is to drop an object that contains a portal, as one would on a terrain region. However, portals are also built into many grid features. 3) You will notice a chunk of floor missing. You must place the cave entrance feature on a spot that has no floor. 4) Select "GreeceNatCave_EntranceG_Int01" from the feature list.

5) Place it in the gap. When you first place the entrance, it may be oriented incorrectly. To rotate it, press the left or right arrow key and click to place it again. The arrow keys rotate the feature placement 90 degrees, but you must place the feature again to see the rotate.

36

8.1.2 Linking Two Portals 1) Select the portal tool.

2) Click the higher ellipsis in the Link Portals Pane.

3) Select the portal at the cave entrance in the cliff on the terrain region. You will not be selecting the object itself, but the actual hole inside of the object.

 You will notice a Portal number appear in the field next to the ellipsis. 4) Click the lower ellipsis in the Link Portals Pane.

5) Click on the plane on the open end of the entrance feature in the cave on the grid region. Rotate the camera so you are inside the cave, looking at the entrance feature.

 37

 You will notice another portal number appear in the field next to the lower ellipsis. 6) Click the 'Link' button in the Link Portals Pane.

 Over each of the portals, the name of the terrain on which they are located will appear. Now the portals are linked. 7) Rebuild All Pathing and Rebuild All Maps for the sake of good habits. 8) Save All. 9) Open up the Art Manager. 10) Find and Build 'Tutorial-Connecting_To_Dungeon.map' in the 'Tutorial08\assets' folder. Now, if you play the map in game, the Portals are connected. Run the character into one of the portals, and the character appears at the linked portal.

8.2 Teleporting

 Another way to move the player around the map is through teleportation. The user can be teleported via a trigger using the 'Teleport Player to N., N., N.)' action in the Quest Editor (Don't worry, I'll explain all about actions in the next tutorial). But how does one obtain the correct coordinates to teleport to?

8.2.1 Obtaining Teleport Coordinates 1) Open up a world in the Editor. 2) Open up a level in Editor Mode.

3) Center the camera on the point you wish to find the teleport coordinates for. 4) Select 'Camera -> Locate Coords' from the Menu Bar. A window displaying the X, Y, and Z coordinates for the point you are looking at will pop up.

 38

 39

Tutorial 9: Making a Trigger

This section introduces the Quest Editor – not to be confused with the Titan Quest Editor, which is used to create maps. Quests are created for the game in quest (".qst") files. Each quest file consists of a series of triggers that control the flow of the quest. Each trigger has a series of conditions and actions such that when all of the conditions for that trigger are met, the actions of that trigger are performed. For this tutorial we will be setting up a simple trigger such that when a player enters an area, he or she gains 10 experience points.

Setup 1) Open the Art Manager 2) Under 'Mod' go to 'Select' and choose Tutorial09 3) Go to the 'Build' menu and select 'Build'   All database records are now accessible by the editors.

9.1 Triggers

9.1.1 Quest Files The trigger we will be creating will be part of a previously existing quest. 1) Open up the Quest Editor. 2) Go to "File -> Select Mod…" and choose "Tutorial09" 3) In the open file window, select and open 'CustomMaps -> Tutorial09 -> source -> Quests -> TutorialExperience_Trigger.qst'. You will see that the only thing in it right now is an entry called "Only One Step". Triggers are grouped into steps to help the designer follow the flow of the mission better.

9.1.2 Creating the Trigger. 1) Select the step you will be adding the trigger to. 2) Click the 'Create New Trigger' button in the toolbar.

40

 A new trigger will appear under the "Only One Step" step. 3) Select the step you just created. 4) You will see a list of properties in the Properties Pane at the base of the screen. To change a property, double click on the cell next to the property name, and type in a new value. 5) Enter the value "The Trigger" into the Name field of the trigger you created.

9.2 Conditions

A trigger is a set of conditions and actions such that when all of the conditions are met, the actions are performed. We'll start by creating the trigger's conditions. In this case, the condition that activates the trigger is when the player enters a specific area.

9.2.1 Creating a Condition 1) Select the Conditions entry under the trigger you wish to add the condition to.

2) Click the 'Create New Condition' button.

This brings up the 'Add New Condition' window. Inside the window is a list of the possible condition types. 3) Select 'Enter Bounding Volume'.

4) Click the 'OK' button. The condition is set set to activate when the player enters a bounding volume, but we still have to specify which bounding volume. 5) Select the condition and look at the Properties Pane. You'll notice it has a property called BaseDatabaseRecord. This is where we establish which bounding volume the player has to enter to activate this trigger. 6) Click on the value field of the BaseDatabaseRecord property.

41

You'll notice that you can type the path in like you normally would on any other property. But you'll also notice there's an ellipsis on the right side of the field that you can click. If you click on this ellipsis it will take the value of whatever asset you have selected in the Asset Chooser Pane on the right. This is very helpful when you don't want to have to type a long, difficult path. 7) In the asset chooser, find and select '<root> -> CustomMaps -> Tutorial09 -> database -> Records -> TutorialSphere_Of_Experience.dbr'. 8) If it's not already selected, select the 'Enter Bounding Volume' condition you just created. 9) In the Properties Pane, select the data field of the BaseDatabaseRecord property, and click on the ellipsis.

The path to the Tutorial-Sphere_Of_Experience will appear on its own. So there we have it. The condition is set so that when the player enters the volume, something will happen. But what's going to happen? For that, we'll need to set up an action.

9.3 Actions

9.3.1 Creating an Action 1) Select the 'Actions' entry under the trigger to which you wish to add the action.

2) Click the 'Create New Action' button.

This brings up the 'Add New Action' window. The window features a list of all the possible action types. 3) Select 'Give Player N, N, N Experience'.

42

4) Press 'OK'. Now the trigger is set up to give the player experience when they enter the area, but we still have to establish how much. 5) Select the new action to view its properties. The three properties we are concerned with are ExperienceEpic, ExperienceLegendary, and ExperienceNormal. These determine how much experience will be given to the player depending on the difficulty setting. 6) You can set them to whatever value you want, but if you need some numbers, I set them all to 100.

9.4 Saving and Building a Quest

9.4.2 Saving and Building the Quest File So now we're done creating the quest. Now we have to save and build it. 1)Save it by selecting 'File -> Save' from the menu bar at the top of the window. Once the quest file is saved, it needs to be built. 2) Open up the Art Manager. 3) Select the Tutorial09 Mod 4) Build it. But wait, we're not done yet!

5) Open the Editor, Choose Tutorial09, and select the .wrl file in the maps folder 6) In the menu bar, go to "Quests -> Select Files…" and choose "Tutorial-Experience_Trigger.qst" 7) Save the level, return to the Art Manager, and build all again.

43

Now we are ready to give this quest a whirl in-game!

9.4.3 Testing Your Trigger. 1) Fire up the Titan Quest game. 2) Select 'Custom Quest' from the main game menu 3) Select 'Tutorial09 ~ Tutorial-Sphere_of_Experience.map' in the Map Pane. 4) Select 'Start' 5) Direct the character into the fancy-tiled circle. If everything has been done correctly, you will see a notification that you have received some experience points.

Tutorial 10: Making a Mod from scratch

So, we've seen how to manipulate previously created .wrl's, .lvl's and .maps to make a playable map for the player. The question becomes, where do .wrl's, .lvl's and .maps come from?

10.1 Make a new Mod

1. Open up the Art Manager.
2. Select 'Mod -> New'. The name of your new mod will be shown in the base folder of all tabs.
3. You will notice that several folders are already in place. For good organization, store all level files under "Maps" and all quest files under "Quests".
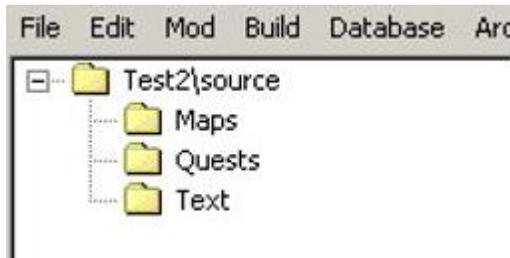
10.2 Create a World and a Level Files

Now you have a place to put your maps. It's time to make them, so run the Editor.

1. When the window comes up, choose your mod name and select the 'Maps' directory
2. Choose a name for your world file, ending in ".lvl" (make sure the "Create a new World" radio button is selected), and click 'OK'.
3. You are now presented with a blank map. You need levels.
4. Select 'Region -> Add New Terrain' (or 'Add New Grid'). Choose a name and a size (NB: 256x256 is big).
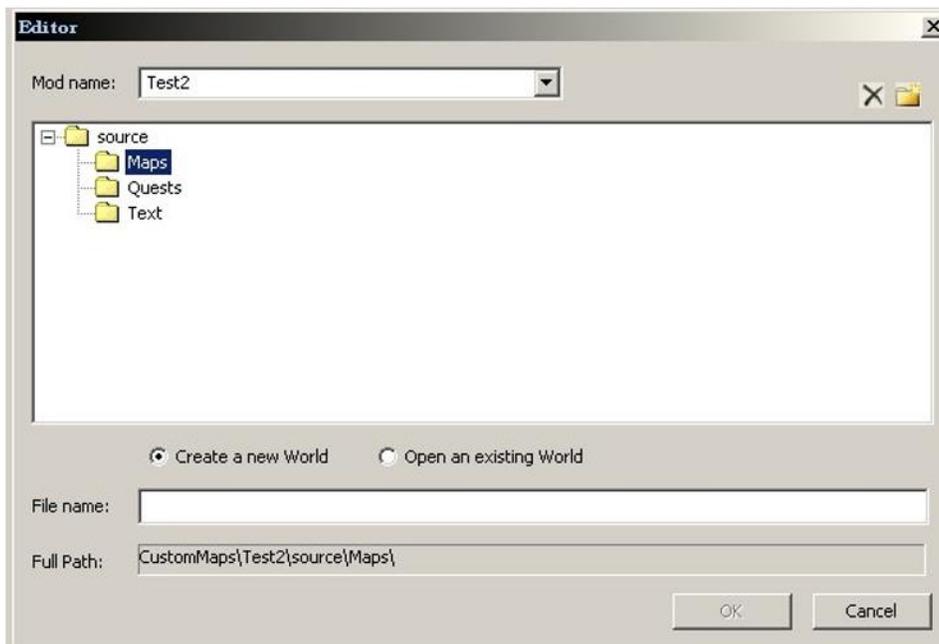
From here, you are ready to terraform, place structures, monsters, and people. However, for the time being, build the pathing and map, save all, and return to the Art Manger.

In the art manager, you should find the files already added to the source tab. Your next task is to create the assets.

10.3 Making Assets

1. Find the '.wrl' file under the 'Sources' tab and right click it and select 'Auto-Create Asset File(s)'
2. The 'Maps' folder will be created a .map file will be created.
3. 3) Build your mod!
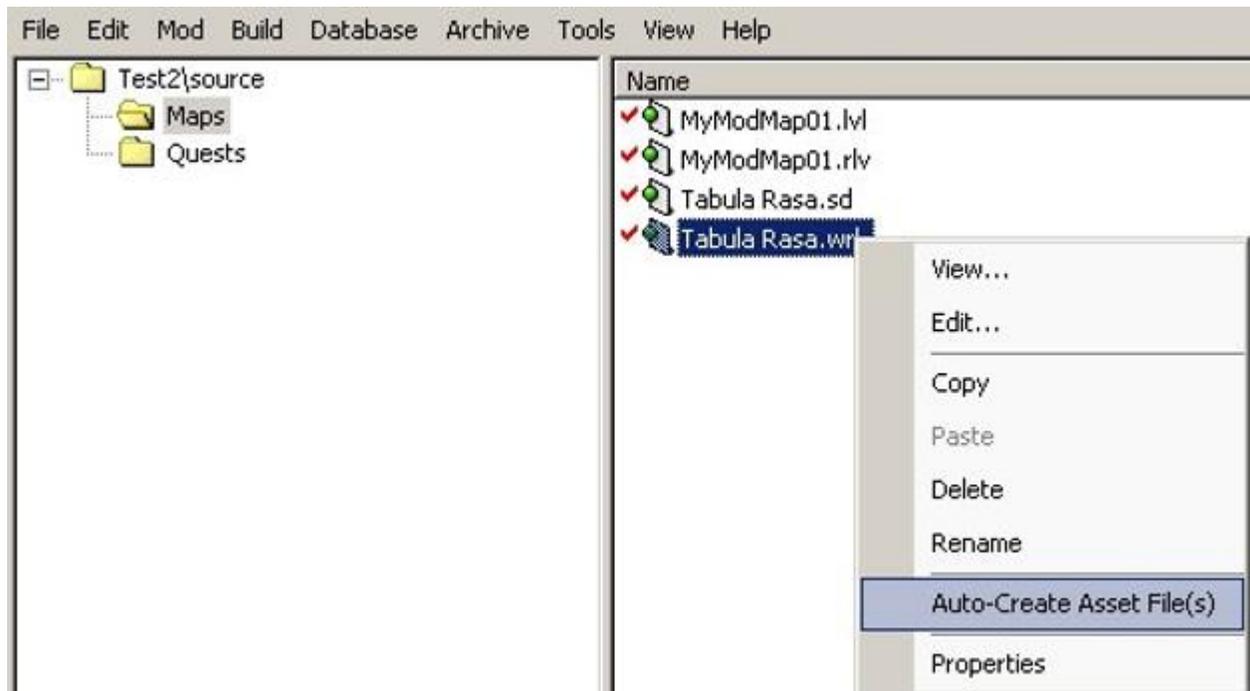
To create and attach quests involves a little more work. See Tutorial 12 for more information on incorporating quests into your mod.

Tutorial 11: Using a New Database Entry

Every object that appears in Titan Quest has a database record. A database record is a series of values that help determine how an object will look and act such as it's 3D mesh, drawscale, and number of hitpoints. For this example we will be making an enormous turtle. This Very Giant Turtle will be identical to the average Giant Turtle, except five times bigger in every direction.

In the Level and Quest Editors, you can see the all the default database records, which you can use to place objects, monsters, and NPCs in the world and tie them into quests. For many quests, you will want to make minor modifications. This will involve importing records into your mod.

11.1.1 Importing Database Records Import default Titan Quest database records to modify or to create new records with. Records you import into your mod and edit will override the default versions – for you mod only.

1) In the Art Manager, select the 'Database' tab and go to 'Mod -> Import Record…' 2) Brows the list of records and select one to import. Navigate to 'Records -> Creature -> Monster -> Giant Turtle -> AM_Snapper_08.dbr'. When you have this record selected, click 'OK'.

The Very Giant Turtle will start off as just a copy of the giant turtle. 3) The imported record will be placed in the same place was found. This will create a lot of extra folders. You can find your newly imported turtle at 'Records -> Creature -> Monster -> Giant Turtle'. For larger mods, you might want to move all your records (using copy and paste) into better organized files in the base folders, such as 'Monsters', 'NPCs', 'Quest' etc.

4) Right click the record and select "Rename" from the context menu 5) Type in the new name of the .dbr. Rename it "Very_Giant_Turtle.dbr".

47

11.1.2 Editing a Database Entry Awesome, so now we have a new object type. But what will differentiate this from any other dumb old turtle? Well, for starters, he'll be about 5 times the size in every direction. To do that we'll need to edit a few of the properties of the Very Giant Turtle. 1) To bring up the Editing Record window, double click on the record you want to edit in the Folder Pane. The Record Editing

Window consists of two panes. On the left is the Hierarchy Pane, which displays the hierarchy of property types for the object being described. On the right is the Properties Pane, where each of the properties and their values are displayed when a category of properties is selected. Each row is a different property and the columns give additional information about the property. The column we are most interested in is the column whose label is the name of the record we're editing, in this case, 'Very_Giant_Turtle.dbr'. Cells in this last column, or value cells, hold the actual value for each of the properties. 2) To change the value, double click on a cell and type in a new value. Many value cells will display an ellipsis when being edited. If the user clicks this ellipsis, that property will take the value of the path of any database record currently selected in the main Art Manager window, thus preventing the user from having to type in a long convoluted file path. We will be editing two properties in the group 'All Groups -> Actor Variables'. The variables are scale, which affects how large the turtle appears to be, and actorRadius, which affects how close another object can get to the center of the Turtle.

3) Select the 'All Groups -> Actor Variables' category in the Hierarchy Pane. 4) Select the value cell of the 'scale' property. 5) Set the value to '5'. 6) Select the value cell of the 'actorRadius' property. 7) Set the value to '5'.

8) Close the Record Editing Window.

11.1.3 Building a Database Entry Now that we've changed the values of the database record, we have to build the new values into the database from which the game reads. 1) Right click on the 'Tutorial11\database' folder. 2) Select 'Build' from the context menu.

  Now that the new database record is built into the database, the user can find it and drop it in the Editor just as they would any other object. So now we know how to make a new object in Titan Quest. Have fun with it, play with the properties and put the objects into your level to see what new fun combinations of properties you can find.

Tutorial 12: Quests

Remember that villager whose village needed rescuing back in Tutorial 3? Well, we're about to delve a little bit farther into how that actually worked. That worked because of a quest file. A quest file, as you may recall, is a series of triggers that control the flow the game. We'll be setting up a series of events that will allow the villager to assign the player a quest, the player to kill the monster, and the villager to reward the player for his work completed. There are 4 major parts to a quest, steps, triggers, conditions, and actions. Steps are a way of grouping triggers together in order to make the flow of the quest easier to understand. Triggers are the real meat and potatoes of the quest. A trigger is a set of conditions and actions such that when all of the conditions are met, the actions are performed.

12.0.1 Setting Up Note: A completed copy of the quest you are about to make is included in this tutorial. Titled 'Sample_TutorialSave_The_Village.qst'. The tutorial guides you through the creation of this file. The created version will be 'TutorialSave_The_Village.qst

The majority of this tutorial takes place in the Quest Editor, but we need to start with the Art Manager. 1) Open the Art Manager 2) Under 'Mod' go to 'Select' and choose Tutorial12 3) Go to the 'Build' menu and select 'Build'. This compiles all database assets. You can ignore any errors regarding quest assets for now.

On to the Quest Editor 3) Open the Quest Editor 4) Go to "File -> Select Mod," and select "Tutorial12". You should always select the mod when creating or editing quest files.

12.1 Tokens Tokens are a way of tracking events that have occurred. When an important quest event happens, the player can be assigned a token that signifies that it happened. Later on in the quest, a trigger can check to see what tokens the player has and determine what action it should take based on what the character has done.

12.1.1 Create a Token In the following example, we will be creating a token that will signify that the player has killed the satyr which can be checked to ensure the quest does not continue before all necessary events have occurred. 1) Click the Token Editor button. You will be prompted for a token file. Select tokens.bin from your mod folder.

2) Click 'Create Token'.

The Add New Token Window will pop up. 3) Enter a name for your token in the text field in the New Token Pane of the Add New Token Window. In this case, name it 'Tutorial_Satyr_Killed'.

 50

4) Click 'OK'. Your token has now been added to the token file. .

12.2 Creating the Actual Quest File

12.2.1 Create a Quest Start by creating a New Quest Document 1) Select 'File -> New Quest Document' from the Menu Bar or click on the New Quest Document icon.

 Set up the quest file's properties. 2) Select the New Quest file in the Files Pane. To change the value of a property in the Property Pane, first select the field on the right, next to the property name. Then type a value into the field. 3) Set 'QuestName' to 'Defeat The Satyr' Rewards are granted to the player when a quest is completed. By setting up reward values in the quest, the player can know what sort of reward they will receive for completing the quest. 4) Set 'RewardGoldEpic' to 200. 5) Set 'RewardGoldLegendary'

to 300. 6) Set 'RewardGoldNormal' to 100. Note: These numbers don't grant the rewards, they just let the quest display what the reward will be. You must use "give X" actions to give the player various rewards.

12.2.2 Create the Steps Steps are a way of grouping together quest triggers by the various events that occur over the course of the quest. Start by creating the first step. 1) Click the 'Create New Quest Step' Button.

2) Select the quest step you just created 3) In the Properties Pane, set the Step's 'Name' property to "Setup" Now create the other three steps in the quest. 4) Click the 'Create New Quest Step' Button. 5) Change the new quest step's 'Name' property to "Talk to the Villager" 6) Create a new quest step. 7) Change the new quest step's 'Name' property to "Kill the Satyr". 8) Create one last quest step 9) Change the new quest step's 'Name' property to "Claim the Reward"

51

12.2.3 Creating a Setup Trigger A setup trigger will help you initialize any settings for any objects in a quest that don't have a desirable default value. In this case, we want the 'Tutorial-Quest_Villager' to have an exclamation point over his head that will signal to the player that he is a primary objective. The trigger will have one condition, and that is the completion of level load. The action that will take place when that condition is met is that it will mark the Villager as a primary objective. This will place the gold exclamation point above the Villager's head. Start by creating the trigger. 1) Select the step "Setup" 2) Click the 'Create New Trigger' button in the toolbar near the top of the screen. 3) Rename the trigger to "Set Up the Villager" Now create the condition. 4) Select the 'Conditions' section underneath the trigger you just created 5) Click the 'Create New Condition' button in the toolbar near the top of the screen. This brings up the 'Add New Condition' window. This window displays all of the types of conditions that can set off actions. 6) Select the 'On Game Level Load' condition. Now, we will be creating the action. The 'Villager' will be marked as the primary objective. 7) Click the 'Create New Action' button near the top of the screen. This brings up the 'Add New Action' window. 8) Select 'Illuminate X as Primary' This action is set to illuminate something as primary. We have to establish what will be designated the primary objective. 9) Select the action you just created. 10) In the Properties Pane, select the right-most field for the action's 'Npc' property. 11) In the Asset Chooser Pane on the right, navigate to and select 'database -> NPCs -> Tutorial-Quest_Villager.dbr'. 12) Click on the ellipsis in the value field for the 'Npc' property of the action you most recently created. Thus completes our first trigger. When the level loads, the villager will be marked as having quest dialog.

12.2.4 Creating an Interaction Trigger. So how does a player get assigned a quest? In this case, they will find out about the quest when they talk to the villager. When the player talks to the villager, the villager will have something to say them, and the quest will be updated in their Quest Log. In the Quest Log, there are two tabs. One is the Summary which lists the mission objectives and describes what is going on in the mission. The other tab is the Dialog tab, which can be used to replay the dialog in which the player is informed of their quest.

52

Start by creating the trigger. 1) Select the second step, "Talk to the Villager". 2) Click the 'Create New Trigger' button near the top of the window. 3) Rename the trigger "Villager Gives Player Quest" Now for the condition. 4) Create a condition for the "Villager Gives Quest to Player" trigger. 5) Select the 'On Interact with X' type condition. The condition is set up so that when the player interacts with something the actions will occur. We now have to set up what the player will interact with. 6) Select the Condition you just created. 7) Set its NPC property to the same 'tutorial-quest_villager.dbr'. Now, when the player interacts with the Villager, something will happen.

8) Add an action to the trigger. Select 'Update X's Dialog to X'.   This action has a special propery: in triggers like this ('On Interact with X -> Update X's Dialog to X'), the dialog will update as you click on them. This NPC had no dialog until now, but when you click on him, we're giving him something to say. Note that at other times, this update dialog action will change an NPCs dialog silently. You'll hear the new dialog the next time you click on them. 9) For the NPC field, add our good old NPC. For the DialogPak field, add 'Database -> dialog -> tutorialvillager_dialog_a.dbr'. 10) We want to add this to the quest log as well, so add another action: 'Update This Quest's Dialog Tab'. Add the same dialog pak here. 11) Add one last action. This time it will be 'Update This Quest's Map Marker'. Here we have something different. The text for the quest objectives is stored in a file called 'ModStrings.txt', which is stored in the Text source folder. You can add entries by putting one 'tag' per line. Each line should look like this: <tag>=<text>.   12) Type Q01Obj into the 'BulletPointTag' field, and Q01Desc into the 'DescriptionTag' field.   13) Set 'DoCompleteBulletPoint' to false – otherwise you'd be completing the quest, instead of assigning it!

53

Time to create the actions involved in updating the player's quest log.

54

8) Create an action for the "Villager Gives Player Quest" trigger. 9) Select the 'Update This Quest's Map Marker' action type. So, the quest's objectives will be updated on the quest log screen. We have to establish what the objective will say, where it will be, and whether it will be marked as complete or not.

An objective's BulletPointTag is the title of the objective as it will appear in the quest objectives screen in-game. 10) Set the action's 'BulletPointTag' property to "Kill The Monster". The objectives screen also displays a little blurb about the current state of the quest. 11) Set the action's 'DescriptionTag' property to "There is a monster attacking the village. Go kill it." We want the objective to be incomplete at this point. 12) Set the action's 'DoCompleteBulletPoint' property to 'false'. In our second trigger, you can talk to the villager in order to receive the quest.

12.2.5 Creating a Creature Death Trigger Now we will be creating a trigger that handles the actual killing of the monster. When the player kills the monster, they receive a token to indicate that they killed the monster. It also updates the objectives in the quest log. Finally, it marks the villager as the new primary objective. 1) Create a trigger in the "Kill The Satyr" quest step. 2) Rename the new trigger "Get Credit for Killing the Monster" 3) Create a new condition in the "Get Credit for Killing the Monster" trigger. 4) Select 'Kill Creature X' from the list of condition types. Establish which monster must be killed for this trigger to go off. 5) Select 'Database -> monsters -> tutorial-quest_satyr.dbr' and transfer it to the 'CreatureDatabaseRecord' field Later on we will need to check if the player has killed the monster yet, so we will have to give the player the token that will signify that he or she has killed the monster. 6) Create a new action in the "Get Credit for Killing the Monster" trigger. 7) Select the 'Give Token X' action type. 8) Select the 'Give Token X' action you just created. 9) Select the ellipsis in the data field of the Token property. 10) Select 'Tutorial_Satyr_Killed' from the Tokens Pane. Next, we should illuminate the Villager as a primary goal, so the player knows to go back to him to finish the quest. 12) Create a new action in the "Get Credit for Killing the Monster" trigger. 13) Select type 'Illuminate X as Primary" from the Add New Action window. 14) Select the new action you just created. 15) Copy in 'tutorial-quest_villager.dbr' Now we need to mark the "Kill The Monster" objective as complete. 16) Create a new action in the "Get Credit for Killing the Monster" trigger of type 'Update This Quest's Map Marker' 17) Set the new action's 'BulletPointTag' property to Q01Obj We should update it so that there is a checkmark next to it. 18) Make sure th enew action's 'DoCompleteBulletPoint' property to 'True' (it should be by default) But we should also add a new objective, telling the player to go back to the Villager. 19) Create a new action in the "Kill the Monster" trigger of type "Update This Quest's Map Marker' 20) Set the new action's 'BulletPointTag' property to Q02Obj And we should update the quest's description blurb. 21) Set the new action's 'DescriptionTag' property to Q02Desc 22) Set the 'DoCompleteBulletPoint' property to 'False'

12.2.6 Finishing the Quest Now the player has to go back and talk to the Villager in order to claim their prize. 1) Create a new trigger in the "Claim the Reward" step. 2) Set this new trigger's 'Name' property to "Talk To Guy To Finish Quest" The first condition of this trigger is that it happens when the player talks to the Villager. 3) Add a new condition to the "Talk To Guy To Finish Quest" trigger of type 'On Interact with X'. 4) Set its NPC property to the same 'tutorial-quest_villager.dbr'. However, we have to make sure first, that the player has killed the monster before giving the player their prize. 5) Add a new condition to the "Talk To Guy To Finish Quest" trigger of type 'Owns Token X'. 6) Set the new condition's 'Token' value to 'Tutorial_Satyr_Killed' So we'll start by changing the Villager's dialog. Now, instead of asking the player to save his village, he'll thank the player for saving it. 7) Add a new action to the "Talk To Guy To

Finish Quest" trigger of type 'Update X's Dialog to X'. 8) Set the new action's 'DialogPak' property to 'Tutorial-Villager_Dialog_B.dbr'.

9) Set the new action's 'NPC' property to 'tutorial-quest_villager.dbr'

 Now we'll change the objective marker to reflect that the player has successfully talked to the Villager. 10) Add a new action to the "Talk to Guy to Finish Quest" trigger of type 'Update This Quest's Map Marker'. 11) Set the new action's 'BulletPointTag' property to "Return To The Villager". 12) Set the new action's 'DoCompleteBulletPoint' property to 'True' As the Villager promised, the player receives their monetary reward. 13) Add a new action to the "Talk To Guy To Finish Quest" trigger of type 'Give Player N,N,N gold'. 14) Select the new action. 15) Set the 'MoneyEpic' property to '200'. 16) Set the 'MoneyLegendary' property to '300'. 17) Set the 'MoneyNormal' property to '100'.

 By marking this quest as completed, we ensure that the quest cannot be completed again by this character. 18) Add a new action to the "Talk to Guy to Finish Quest" trigger of type 'Complete Quest Now'. 19) Set the new action's 'QuestFile' property to 'Quests\Tutorial-Save_The_Village.qst'.

12.3 Making Objectives Appear At this point, if you were to build everything and go in-game, the quest would technically function. When you killed the monster, the villager's dialogue would change. When you went and talked to the villager the second time, you would receive all of the rewards for completing the quest. Unfortunately, none of the mission objectives we created would appear in your mission log and there would be no indication on the map as to where the player was. In order for these to happen, the player must create a map marker and overlay. A map marker consists of a few parts. The map marker itself, is a ring that shows up on the map in the Quest Log when the player checks their objective. The overlay, is the gem that appears inside the map marker to show that the player is in that zone. Lastly, the map marker must be linked into a quest location map.

First, we want to import a few files from Titan Quest, as templates to work off of: 1) Open the Art Manager 2) Click the Database Tab. 3) Go to 'Mod -> Import Record…' 4) Navigate to 'Records -> InGameUI -> Player Quests -> Map Markers' and import 'GreeceQ1.dbr' and 'GreeceQ1_overlay.dbr' (you'll have to import each separately).   Then import 'Records -> InGameUI -> Player Quests -> QuestLocationMap1.dbr'. 5) Copy the files and paste them into the Quests database folder (right click in the Folder Pane) – and rename them to 'Tutorial-Save_The_Village.dbr', 'Tutorial-Save_The_Village_overlay.dbr', and 'Tutorial-QuestLocationMap.dbr'.

 1) Open 'Tutorial-Save_The_Village.dbr' to edit.   2) Select 'Config' from the pane on the left. 3) Select the 'marekerButtonOverlay' field in the fourth cell from the left and double-click the cell

4) Click the '…' button to choose the file, and select 'quest -> tutorial-save_the_village_overlay.dbr' 5) Change the following cells so they match the screenshot:

The 'MarkerBitmap' fields can stay the same – they are set to use the same map marker art as Titan Quest.

12.3.3 Creating a Quest Location Map

We're almost there!

1) Edit Tutorial-QuestLocationMap.dbr 2) Under 'Config', edit the 'mapMarkers' cell. A box will come up listing a number of fields. Each one is a quest. Delete all but one. 3) Double-click the remaining records value and click the '…' button to locate our quest map marker 4) Navigate to 'Records\Quest\Tutorial-Save_The_Village_overlay.dbr'

5) Press the 'OK' button, close the record and save it. 6) You're all set. Build your mod.

12.4 Putting Quests Into Maps

12.4.1 Placing Quest Objects This may ring a few bells, but here's a quick reminder.

First, associate the quest 1) Open the Editor and select the mod Tutorial12 56

2) Find and open the 'Tutorial – Save_The_Village.wrl' 3) Go the the Quest menu and click on 'Select Files...'. Choose your quest file.

Next, we want to place all the quest objects: 1) Open 'Tutorial-Save_The_Village_Terrain.lvl' in Editor Mode. 2) Select the Object Placement Tool. Drop the special villager 3) Place a 'NPCs -> Tutorial-Quest_Villager' on the terrain. Drop the special satyr 4) Place a 'Monsters -> Tutorial-Quest_Satyr' on the terrain. Place a spawn point. 5) Drop a 'records -> controls -> spawnplayer' on the terrain.

Finally: 1) Enter Layout Mode. 2) Select 'Build -> Rebuild All Maps' from the Menu Bar. 3) Select 'Build -> Rebuild All Pathing' from the Menu Bar. 4) Select 'File -> Save All' from the Menu Bar.

12.5 Building and Testing

12.5.1 Building Everything 1) Open up the Art Manager. 2) Select 'Build -> Build' from the Menu Bar. All of the assets for Tutorial12 will build. 3) Run Titan Quest, select 'Play Custom Quest' and locate your new map.

 And there you have it. When you talk to the villager, the objectives will appear in the quest screen. When you kill the monster, the objectives will update again. When you talk to the villager after killing the monster, the quest will complete and you will be rewarded.

Tutorial 13: Group

This tool is used to link objects together. First one creates a group and each group is given a name and a type.

Proxy Group Types

Monsters in the world are spawned with Proxies and by putting a proxy in a group you can change its behavior. It can also be used to spawn proxy shrines that boost the player.

- Default: If the proxy is in no group it will act like default.
- Unified Proxies: Make all the proxies in the same group use the same pool.
- Unique Proxies: When a Unique Proxies group is used only one proxy will spawn from the group.

Wanderers and Patrollers

To make the world a bit more lifelike it is possible to have NPC's (wanderers) or monsters (patrollers) that walk around in the world. The setup is the same for both but they use different group types. NPC's that have the idleBetweenWaypoints set in the DBR will pause at each waypoint.

1. Create the waypoints. If you make waypoints for NPC's use the classtype NpcWanderPoint and for monsters use the classtype PatrolPoint in the dbr's you place out.
2. Group the waypoints. Use the group type Wander Points or Patrol Points.
3. Group the NPC's or monsters. Put the mobs in the NPC Wanderers or ProxyPatrollers group type.
4. Link the waypoint group and the mob group. Select the mob group, select Link mode and then click the black dot for the waypoint group.

Shrine's

All respawn shrines in an act should be in the same group. All teleport shrines in an act should be in the same group.

Any Entity

Bandari

Bandari Teleport Points

Decorations

Monsters And Weapons

Patrollers

Terracotta Warriros