

## **Making Your World Fleshy** – ScriptEase performances and behaviours

The goal of this tutorial is to build on the last tutorial (Making Your World with Triggers) by further fleshing out your world and making it seem more alive. This tutorial assumes that you've finished the previous five Make Your World tutorials.

In this tutorial, we will explore ScriptEase behaviours that outline simple actions for NPCs to make it seem like they are interacting with the world. Having other folk around, going about their business, further immerses the player in your world. We will create a farmer and his chickens.

The player is a wandering adventurer who happens upon a tower in the wilderness. Inside the tower, the player finds a friendly Guardian who tells the player about the magic levers. If they are pulled in the right order, the vault opens and the treasure may be obtained.

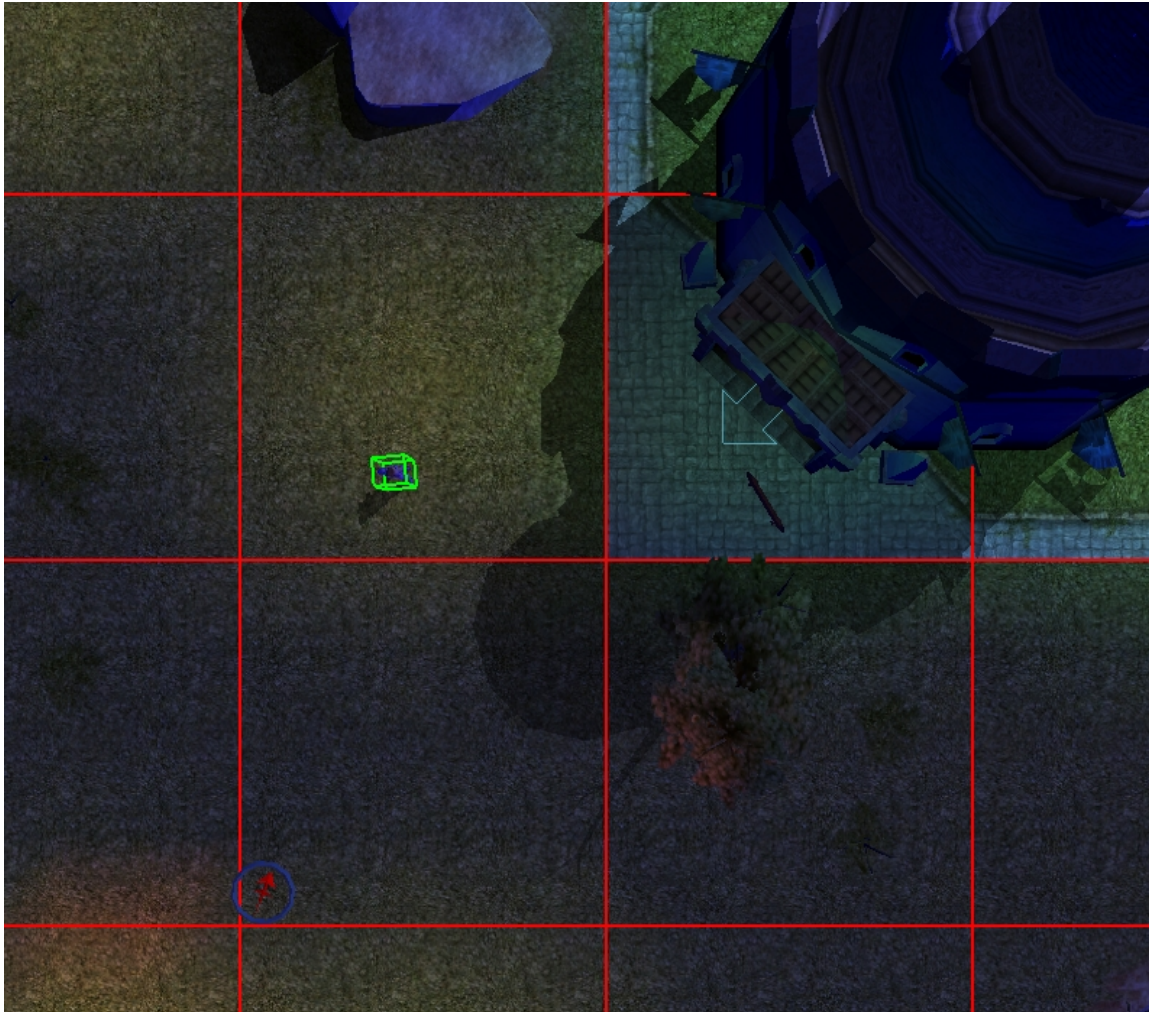
### **Getting Started:**

- 1) Open The Aurora Toolset.
- 2) Open LeverMagic5 and save it as "LeverMagic6".

### **Preparing the World:**

Finding the Farmer:

- 3) Using our (hopefully) familiar technique of "Edit Copy", make an NPC → Human → Commoner, Male. In the "Creature Properties" window, change "First Name" to "Ovar", the "Last Name" to "Reeacktan", and the "Tag" to "ovar".
- 4) Paint Ovar in the "Countryside" area, somewhere near the entrance, but off to the side a little so that it doesn't look like he's...well...hanging around the entrance (though he will later). Rotate him so that he's facing toward where the PC will spawn (aka the "Start Location"). Speaking of the start location, let's move it back from the entrance a bit, to encourage the PC to look around a bit before charging into the tower. Use the "Paint Start Location" tool to paint a new one or just click on the existing starting location and drag it back a bit. Here is my area viewer with the new start location and Ovar:



### **Ovar Learns to Talk:**

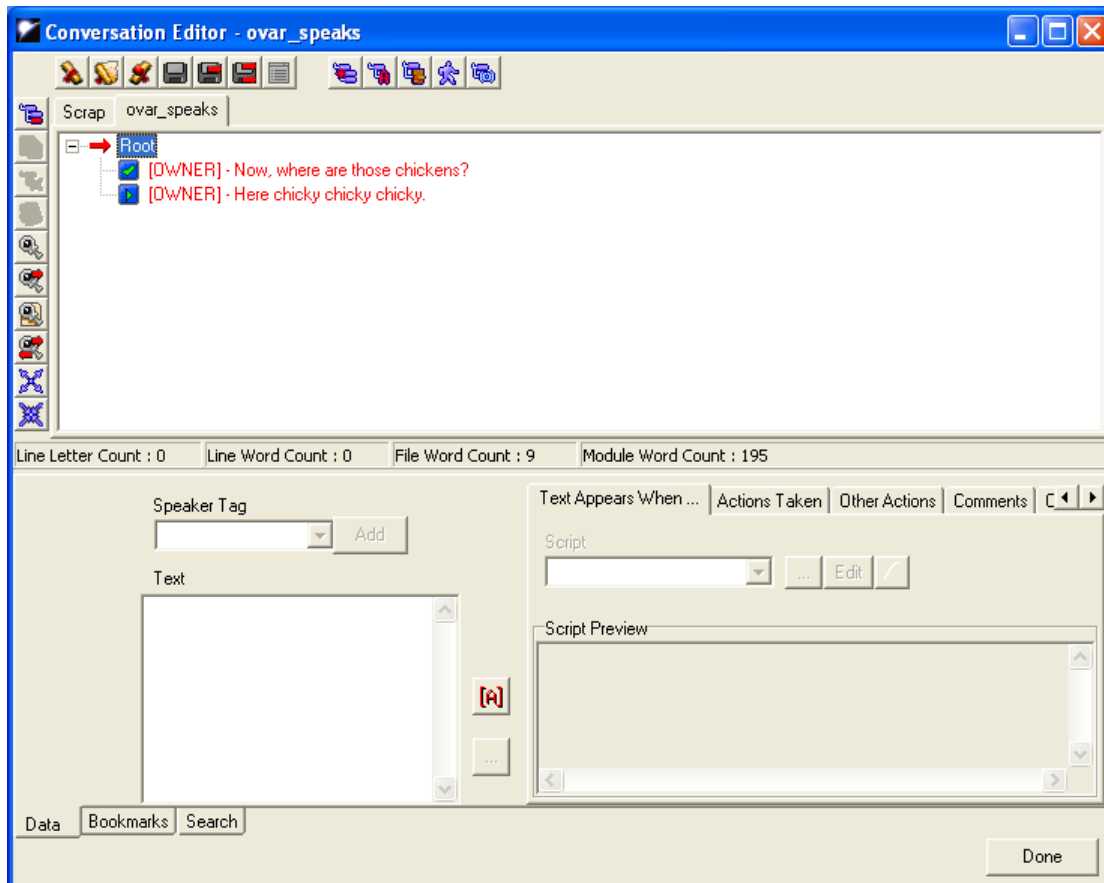
5) Behaviours often include one-liners that the actors speak as they go about their business. These one-liners are selected from conversation files. Create a new conversation file (Under the “Tools” menu in the main menu bar, choose “Conversation Editor”). Click the “Add” button or right-click on the “Root” of the conversation and choose “Add” from the menu.

6) In the “Input Text” window that opens, type “Now, where are those chickens?” and click the “OK” button to add the one-liner.

7) To make more interesting NPCs, we can make them randomly choose one of several one-liners. To do that, let’s add another choice.

Add another line to the **root**:  
“Here chicky chicky chicky.”

8) Click the “Save” button, and call your conversation “ovar\_speaks”.



9) Close “ovar\_speaks” and create a new conversation by clicking the “create a new file” button. Add the line “I am tired of this.” and save the new conversation as “ovar\_returns”.

### **Ovar Behaves:**

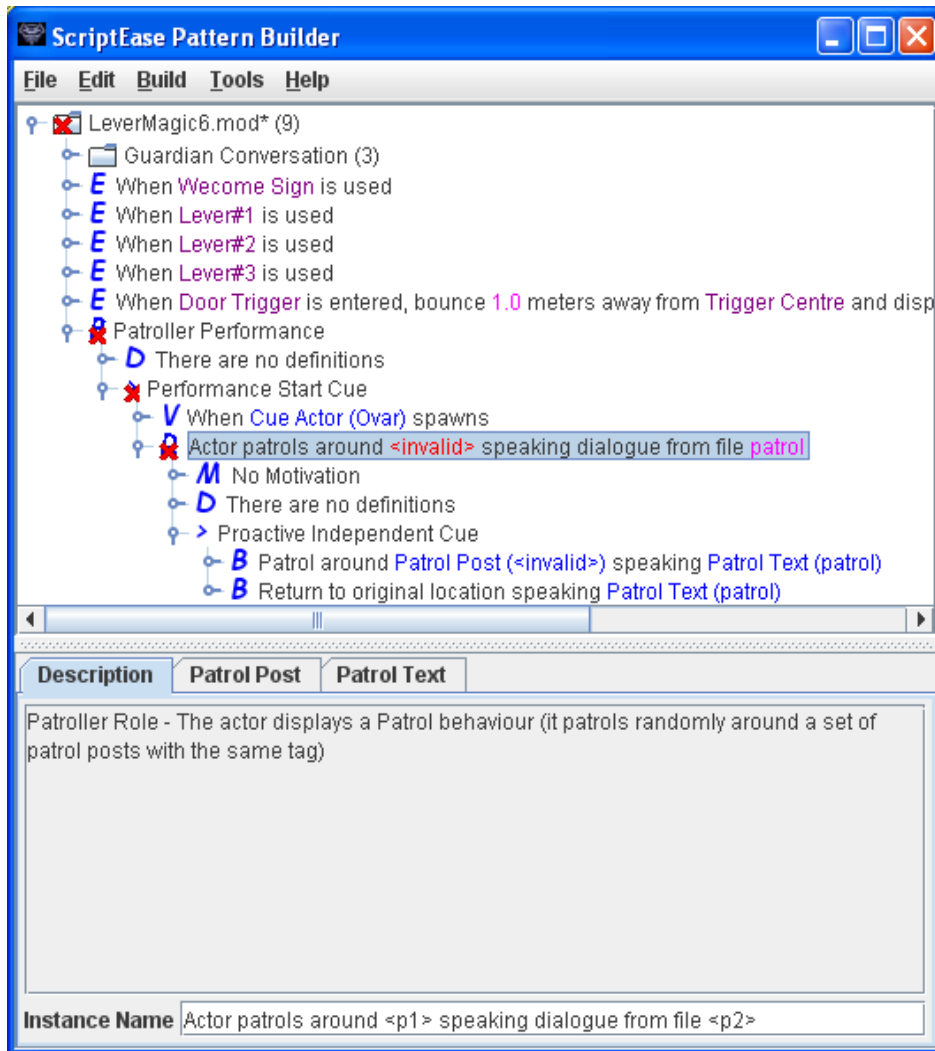
10) Save your module. Close it in Aurora and open it in SE. Right-click LeverMagic6.mod to bring up the pop-up menu. Choose New Specific Performance → Prebuilt Performances → Patroller Performance

As you can see, there are lots of pre-built performances to choose from. We use the “Patroller” Performance. This gives us a character that patrols around a post.

11) Every Performance needs someone to perform. This “someone” is selected in the “Actor” tab. Make sure the Patroller Performance is selected, and click the Actor tab. Under the “Actor” tab, choose Module Blueprint → Pick..., and pick Ovar from the drop-down menu.

12) Inside the Patroller Performance, select the Patroller Role (the line starting with a

blue R, blocked by a red x currently, see image below). Note that inside the Patroller Role, there are behaviours (indicated by a blue B).



13) Under the “Patrol Post” tab, we’ll select the point Ovar will patrol to. He should move to his post. You can use almost anything for a post point, but we’ll use the “Welcome Sign” because it’s already there. “Pick...” the sign from the “Placeables” section of the “Pick a blueprint” window.

14) Under the “Patrol Text” tab, type “ovar\_speaks” in the “Constant” field. This will be the conversation Ovar will speak when he performs the animation. Remember “ovar\_speaks” is the name of the conversation file created in the Toolset. For a small check, at this point, the red ‘X’ indicating an incomplete Role should be gone. If it’s not, re-read the last few steps to see if you missed anything.

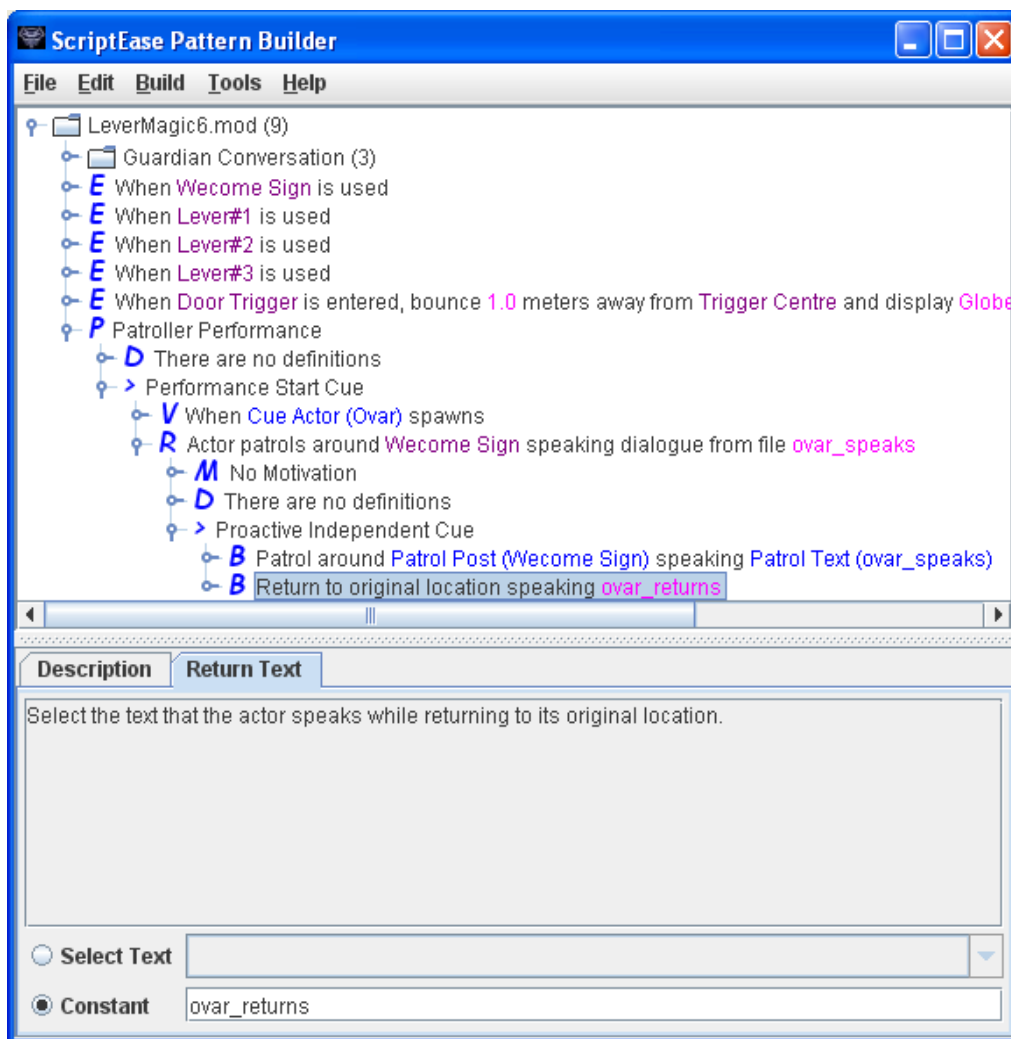
15) Now Ovar will only speak the first line from the conversation “ovar\_speaks”. Let’s make him say something different when he returns to his original spot.

Expand the Role and the Proactive Independent Cue, you'll see two Behaviours, Patrol and Return. Select the Behaviour "Return to original location..."

Under the "Return Text" tab → Constant → type "ovar\_returns".

**Note:** for behaviours, you must enter the name of a conversation created in Aurora, you cannot just type in a sentence here!

16) Save and compile your module then play it to see how Ovar reacts. He should look around for a bit, walk to the sign, look around for a bit, and walk back, jabbering all the while.



17) Remember that we've created two lines in the conversation "ovar\_speaks"? Right now he only ever speaks the first line. How do we make him randomly choose one? Easy. Right-click LeverMagic6.mod → New Specific Encounter → Encounters → Base CodePak → Conversations → Randomize Conversation

18) Under the “Conversation” tab, Pick “ovar\_speaks”.

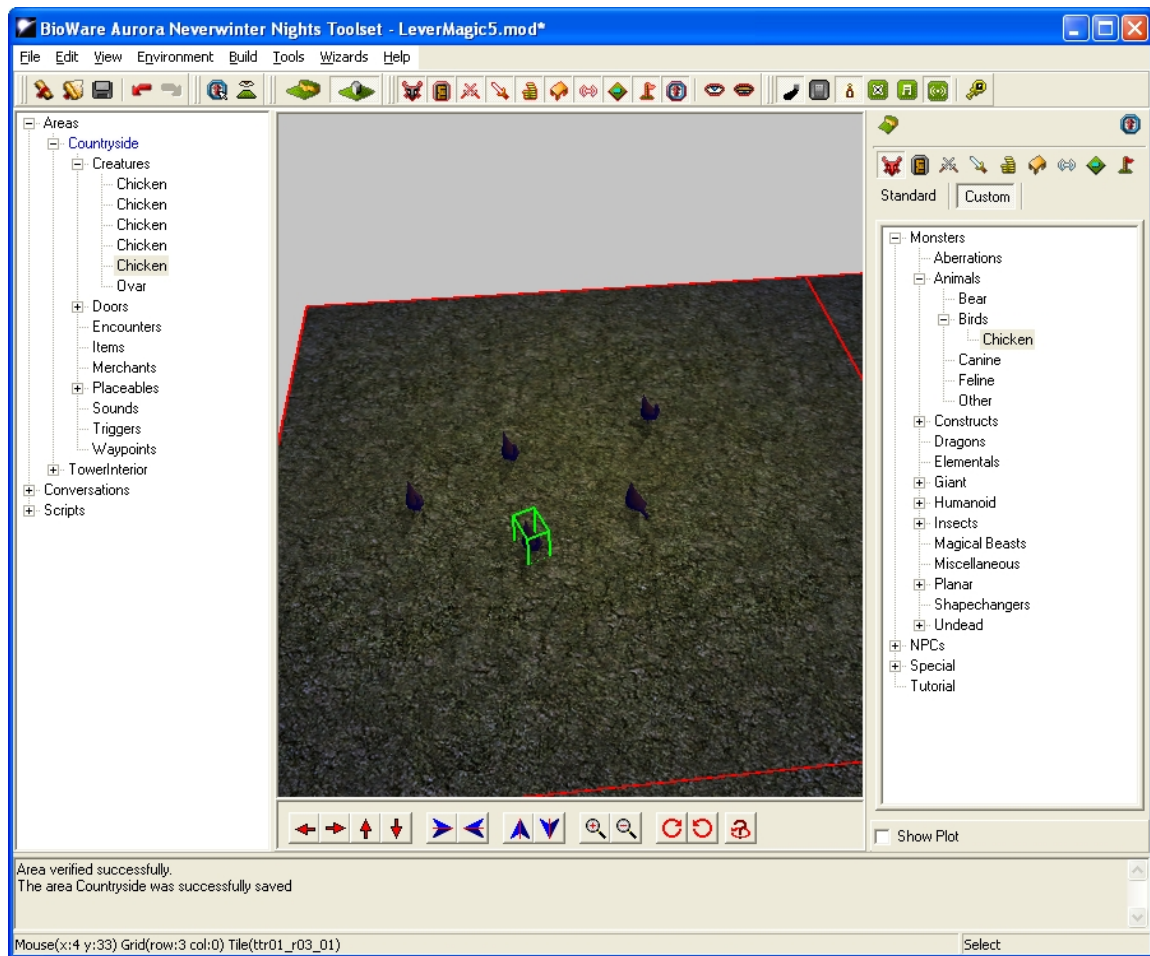
19) Save and compile. Watch Ovar for a while and you’ll see that he is now randomly choosing between the two lines in “ovar\_speaks”. This works for multiple lines too.

### **The Chickens!**

20) Close your module in SE and open it in the toolset. Open the “Countryside” area.

21) Using the “Edit Copy” technique, create a chicken (Paint Creatures → Monsters → Animals → Birds → Chicken) and change the Tag to “ovar\_chicken”.

22) Pick a corner of the map (away from Ovar) and paint a few copies of your new chicken. To paint multiple copies, paint your first chicken the way you normally would, then, under the “Edit” menu in the main menu bar, choose “Copy”. Then, under the same menu, choose “Paste”. When you put your mouse cursor back in the area viewer, it will now have a new chicken attached to it. Just click the ground near your other chickens to place it. I painted five chickens, but it really doesn’t matter how many you paint.



23) Create a conversation for the chickens much like you did for Ovar. Create a conversation named “chickens\_speak” with the line “Bock! Bock! Bock!”.

24) Just for giggles, we’ll create another conversation for the chickens called “chickens\_warn”. Include the line “Bock off!” (which is not chicken profanity, but just chickenese for “Back off”). When you are finished with your chicken talk, save your module, close it, and open it in ScriptEase.

### **Building your own performance:**

For Ovar, we’ve used a prebuilt Performance and made him a Patroller. For the chickens, let’s see how we can build a new performance from scratch. It’s not hard actually.

25) Right-click LeverMagic6.mod to bring up the pop-up menu. Choose New Specific Performance → New Performance.

26) In the New Performance, under the “Description” tab, change the Pattern Name to “Chicken Warning Performance”. Maximize the window if you can’t find the textbox. Under the “Actor” tab, Module Blueprint → Pick... → choose “Chicken” from the dropdown.

The cool thing here is that because the tags for all of our chickens are the same, this performance will apply to all of our chickens!

27) Since this is a new performance, it contains no behaviours, which means the chickens won’t do anything. Not yet. Let’s make the chickens run around in the field.

Under the Chicken Warning Performance, there is a Cue called “Performance Start Cue”. A cue (just like a theatrical cue for actors in a movie) tells an actor (NPC) when to carry out the behaviours inside this cue. “Performance Start” means that the behaviours inside this cue should be carried out as soon as the performance starts, which is at the start of the game.

Leave the Performance Start Cue the way it is. Let’s create a new Cue that makes the actor do things at certain time of the day. Such a cue is called a Timer Cue.

Right-click “Chicken Warning Performance”, New Role Cue Instance → Timer Cue.

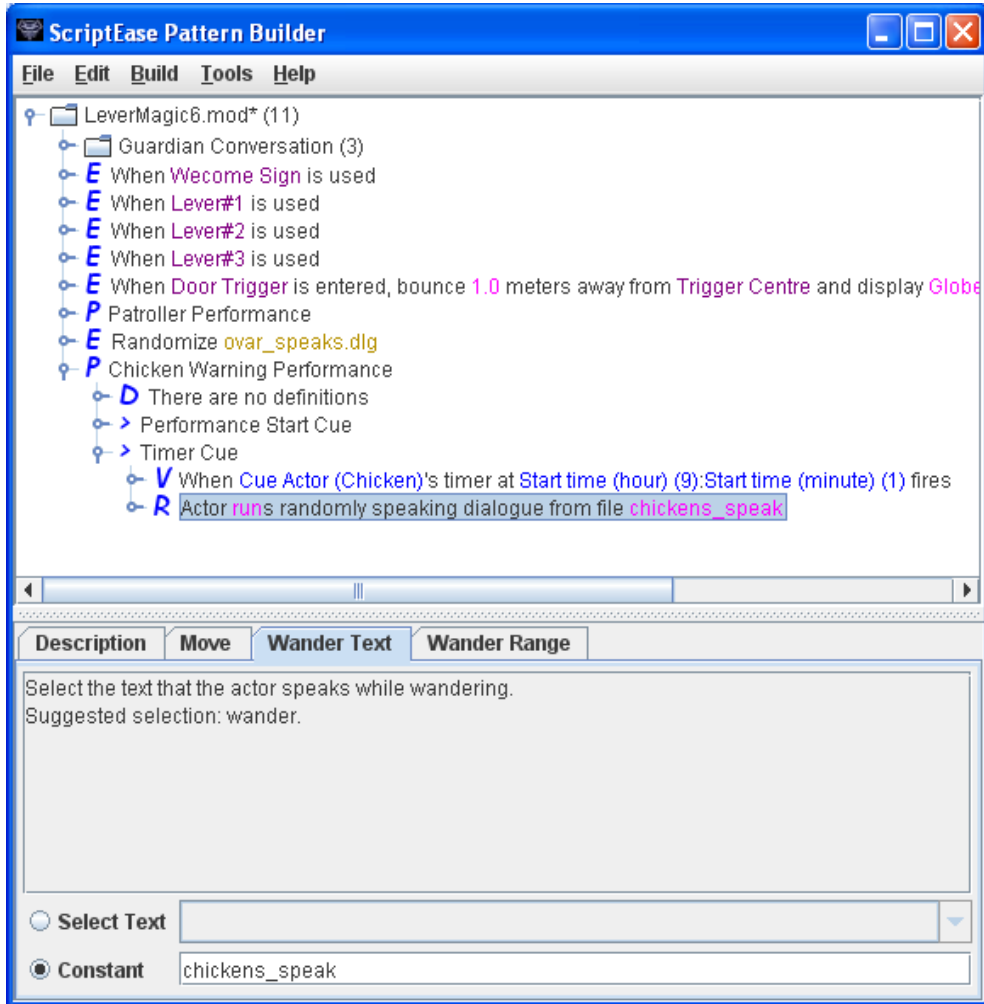
28) In the Timer Cue, under the “Start time (hour)” tab, change Constant to 9. Under the “Start time (minute)” tab, change Constant to 1. This means that the actor will take on whatever role and its behaviours inside this Cue, when the in-game time reaches 9:01.

**Note:** By default, in Neverwinter Nights, one hour has only 2 minutes! So 9:02 = 10:00. You can change this in your game, by opening your module in the Toolset, and go to the Edit menu → Module Properties → Advanced tab.

Let’s leave that for now.

29) Right-click the Timer Cue, New Specific Role → Actor ([run](#) / [walk](#))s randomly speaking dialogue from file [Wander Text](#).

30) In the new wanderer Role we just created, under the “Move” tab, select Constant → run. Under the “Wander Text” tab, change the text in the Constant textbox to “chickens\_speak”. This is the name of the dialogue file we created. Under the “Wander Range” tab, change the Constant to 3.0. Check the screenshot below.



You can test your module now. The chickens should be running around aimlessly.

Now, what if we want the chickens to change their behaviours at another time? For example, in the afternoon, we want the chickens to become paranoid. (I know, this does not make sense at all, but this is a game, you know...)

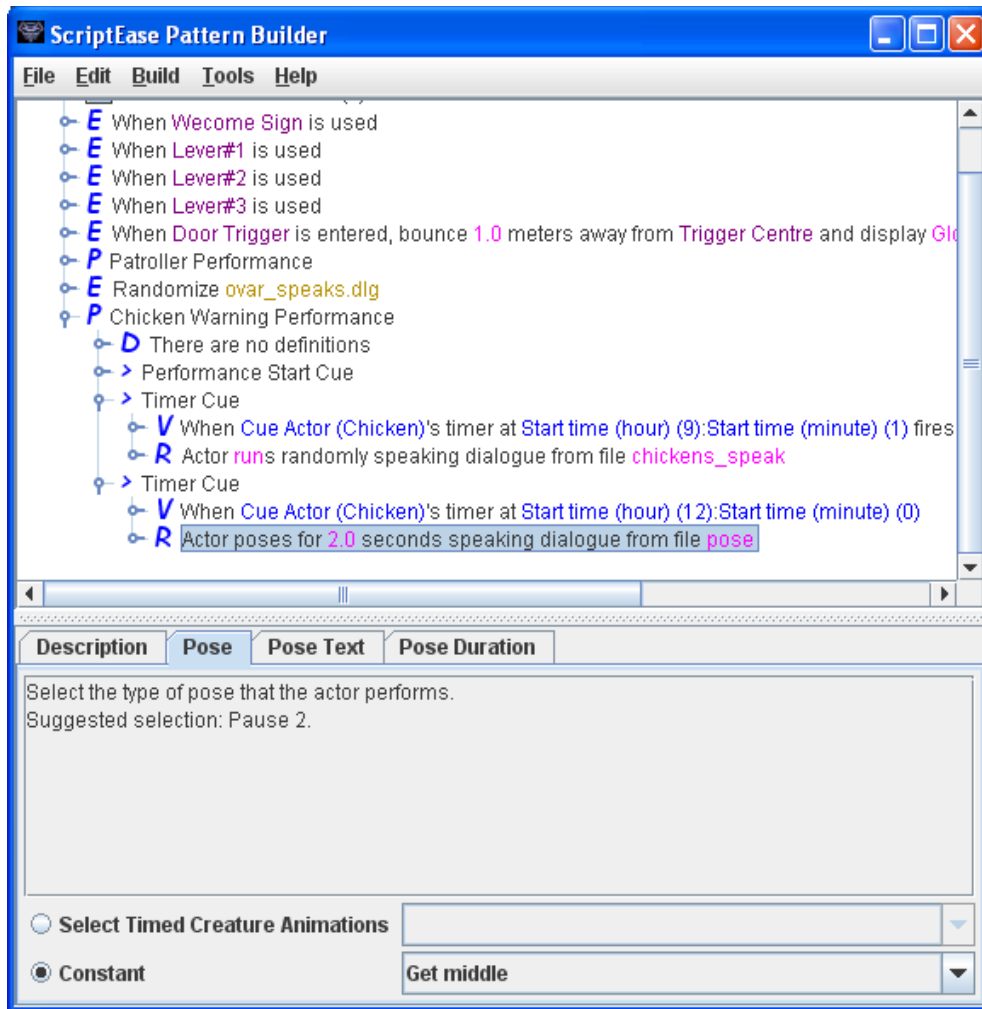
31) For this, we need another Timer Cue. Right-click the Chicken Warning Performance, New Role Cue Instance → Timer Cue.

32) In the Timer Cue, under the “Start time (hour)” tab, change Constant to 12. Under the “Start time (minute)” tab, change Constant to 0. This represents 12:00 at noon. It’s a 24-hour clock.

33) Right-click this new Timer Cue, New Specific Role → Actor poses for [Pose Duration](#) seconds speaking dialogue from file [Pose Text](#).

34) Under the “Pose” tab, choose the “Get middle” animation from the “Constant”

dropdown. This makes the chickens jump. This is a good time to note that many models (like chickens, for example) have only a few animations. Chickens can't conjure or act drunk. They can really only eat and jump. Check the screenshot below.



35) Lastly, we want the chickens to warn the player character when he gets too close. We still want the chickens to jump when the PC is not around, so we don't want to create a third Cue under the Performance. An actor can only be in one Role at a time, so we create a Cue inside the Role. To check distance, we use a Range Cue.

Right-click the Role "Actor poses for 2.0 seconds speaking dialogue from file pose", New Behaviour Cue Instance → Range Cue.

**Note:** A Cue inside a Performance is called a Role Cue.  
A Cue inside a Role is called a Behaviour Cue.  
Role Cues contain Roles. Behaviour Cues contain Behaviours.

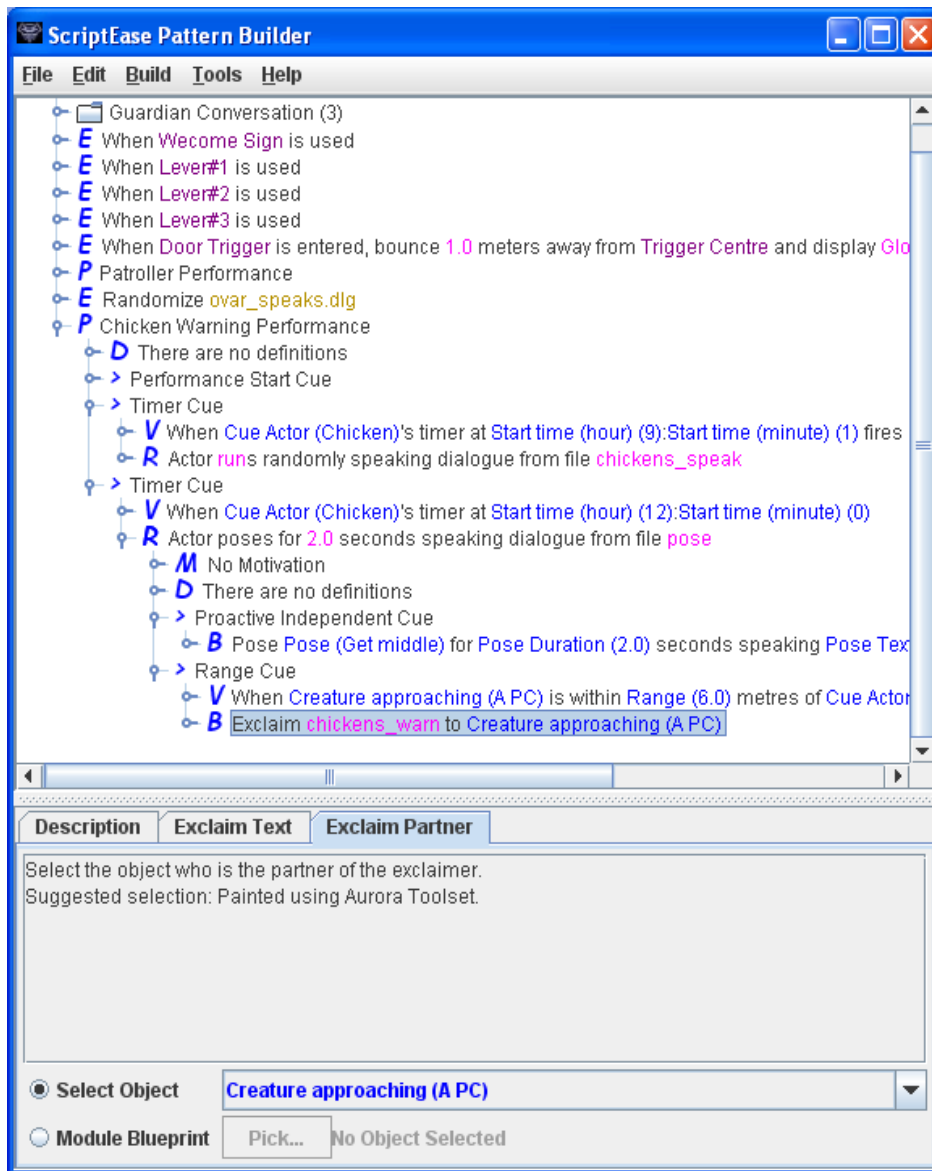
36) For this Range Cue, under the "Creature approaching" tab, Module Blueprint →

Pick... → Select a PC.

37) Now we add a Behaviour inside this Range Cue. Right-click the Range Cue, New Basic Behaviour Instance → Exclaim [Exclaim Text](#) to [Exclaim Partner](#).

38) For this new exclaim Behaviour, under the “Exclaim Text” tab, enter “chickens\_warn” in the Constant textbox. Under the “Exclaim Partner” tab, dropdown menu → [Creature approaching \(A PC\)](#).

39) Save and compile your module and check it out in NWN. Between 9:01 and 12:00, the chickens should be bocking. Between 12:00 and 9:01 of the next day, the chickens should be jumping and should warn you away when you try to approach them. The completed module is shown below.



Just a side note, you can specify the starting time of your module by going into Module Properties in the Toolset, in the Advanced tab. In game, you can check the current time by pointing your mouse cursor at the compass, as shown below:

