Making Your World Leverly -- plot tokens and other magic

The goal of this tutorial is to build on the last tutorial (Making Your World Interesting) by making the world you've built even better. This tutorial assumes that you've finished the basic Aurora Toolset tutorial, the ScriptEase tutorial, and the previous three Make Your World tutorials.

In this tutorial, we will get the lever animation down, create this "right order" we've been babbling about, and make some treasure!

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 LeverMagic3 and save it as “LeverMagic4”.

Animating Placeables:

Levers don't play nice:

3) You may have noticed that we've been avoiding the levers. Only one even moves, it flips on and off really fast, and it doesn't affect the world at all. You may have surmised that this is because levers don't play nice. You may have surmised correctly. Let's get started, and you’ll see why levers are tricky as we go. Close your module in Aurora and open it in ScriptEase.

4) Expand the “Lever #1 use” Encounter and the “Use placeable...” Situation.

5) First, we'll delete that “Action” that is trying to animate the lever by right-clicking on the “Then, Animate...” Action, and choosing “Delete” from the menu.

6) Next, we'll Add a Definition-->Get Variables-->Define a local integer to the “Use placeable...” Situation. Because the lever has no sense of whether it is in an “on” (the game thinks of it as “activated”) or “off” (“deactivated”) state, we must keep track of this ourselves. We do this with a local variable which will change (to reflect the change in the lever's state) each time the lever is pulled. For example, when the local integer is “1”, that means the lever is in the “activated” or “on” state. When the player “uses” the lever in the game, the local integer gets changed to “0”, and the lever changes state accordingly. If this doesn't make sense, don't worry. It will become clearer as we go along.

7) In the “Definition” you've just created-->Description tab-->Label-->put “LeverToggle” into the textbox.

8) Under the “Target” tab-->Select Object-->The Placeable. As you've probably guessed, we can simply choose “Lever #1” from the Module Blueprint-->Pick option. There is a reason for choosing the “variable” name over the “constant” name, but we'll go into that when we start animating the other levers.

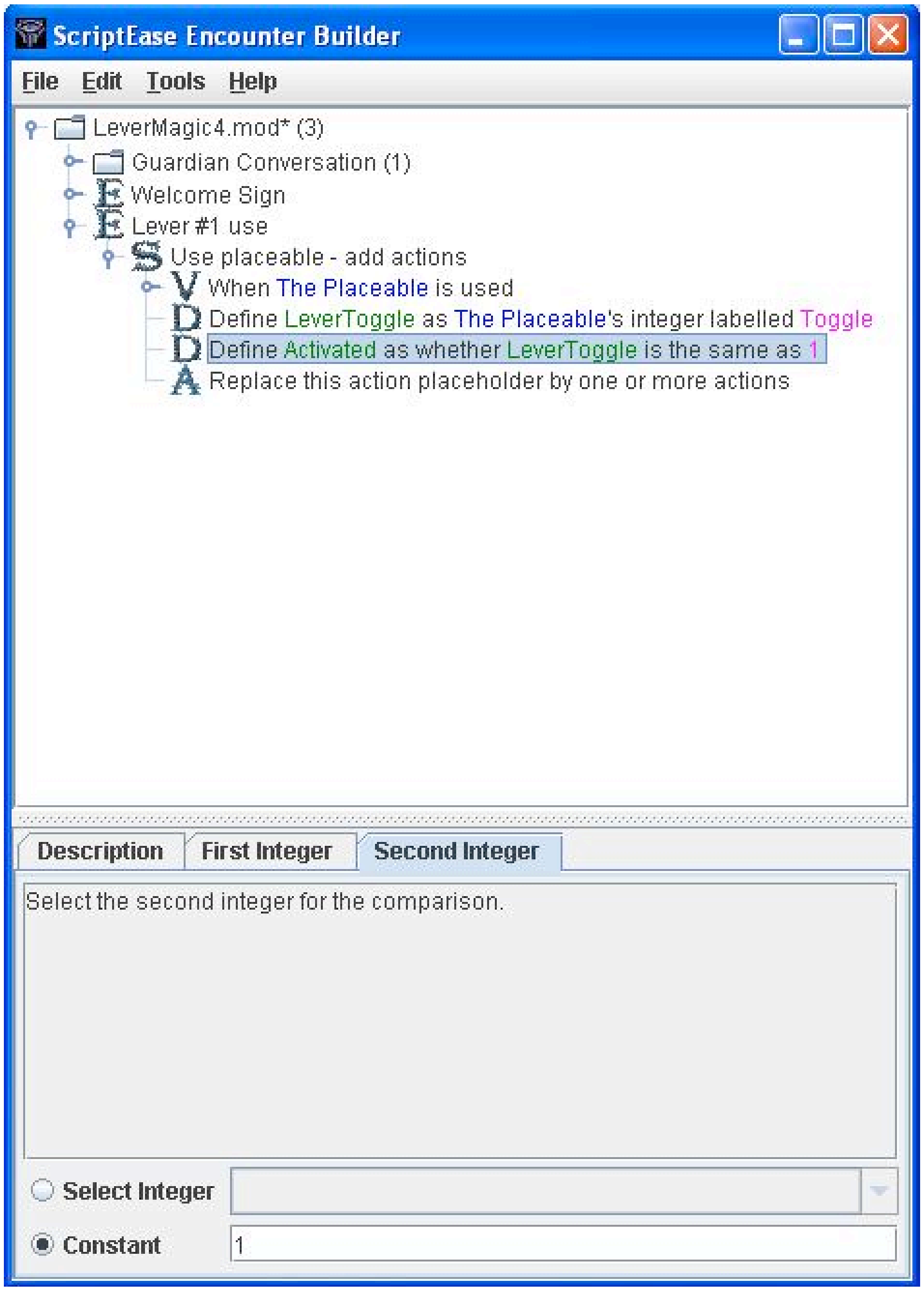
9) Under the “Label” tab-->Constant-->put “toggle” into the textbox.

10) Again, under the “Use placeable...” Situation-->Add a Definition-->Binary-->Comparing Objects or Values-->Whether two integers are the same.

11) Under the “Description” tab-->Label-->put “Activated” in the text box.

12) Under the “First Integer” tab-->Select Integer-->choose “LeverToggle” from the dropdown.

13) Under the “Second Integer” tab-->Constant-->put “1” in the text box. At this point, your module should look like the screenshot below.


ScriptEase Encounter Builder

File Edit Tools Help

```
LeverMagic4.mod* (3)
    Guardian Conversation (1)
    Welcome Sign
    Lever #1 use
        Use placeable - add actions
            When The Placeable is used
            Define LeverToggle as The Placeable's integer labelled Toggle
            Define Activated as whether LeverToggle is the same as 1
            Replace this action placeholder by one or more actions
```

Description | First Integer | Second Integer

Select the second integer for the comparison.

Select Integer

[✓ Constant] 1

14) Now we'll Add an Action-->Conditional Action-->Do the following actions if true to the "Use placeable..." Situation.

15) Under the "Condition" tab-->Select Binary-->choose "Activated" from the dropdown.

16) Now repeat steps 14 and 15, but choose "Do the following actions if FALSE". This is to catch both situations: when the lever is "Activated" and when it is "Deactivated".

17) Under the true “Action”, we need to toggle the local variable's state (from 1 to 0) and we need to animate the lever. First we'll toggle by Add an Action-->Action Atom-->Variable Assignment-->Assign a value to a local integer variable.

18) Under the “Target” tab-->Select Object-->The Placeable.

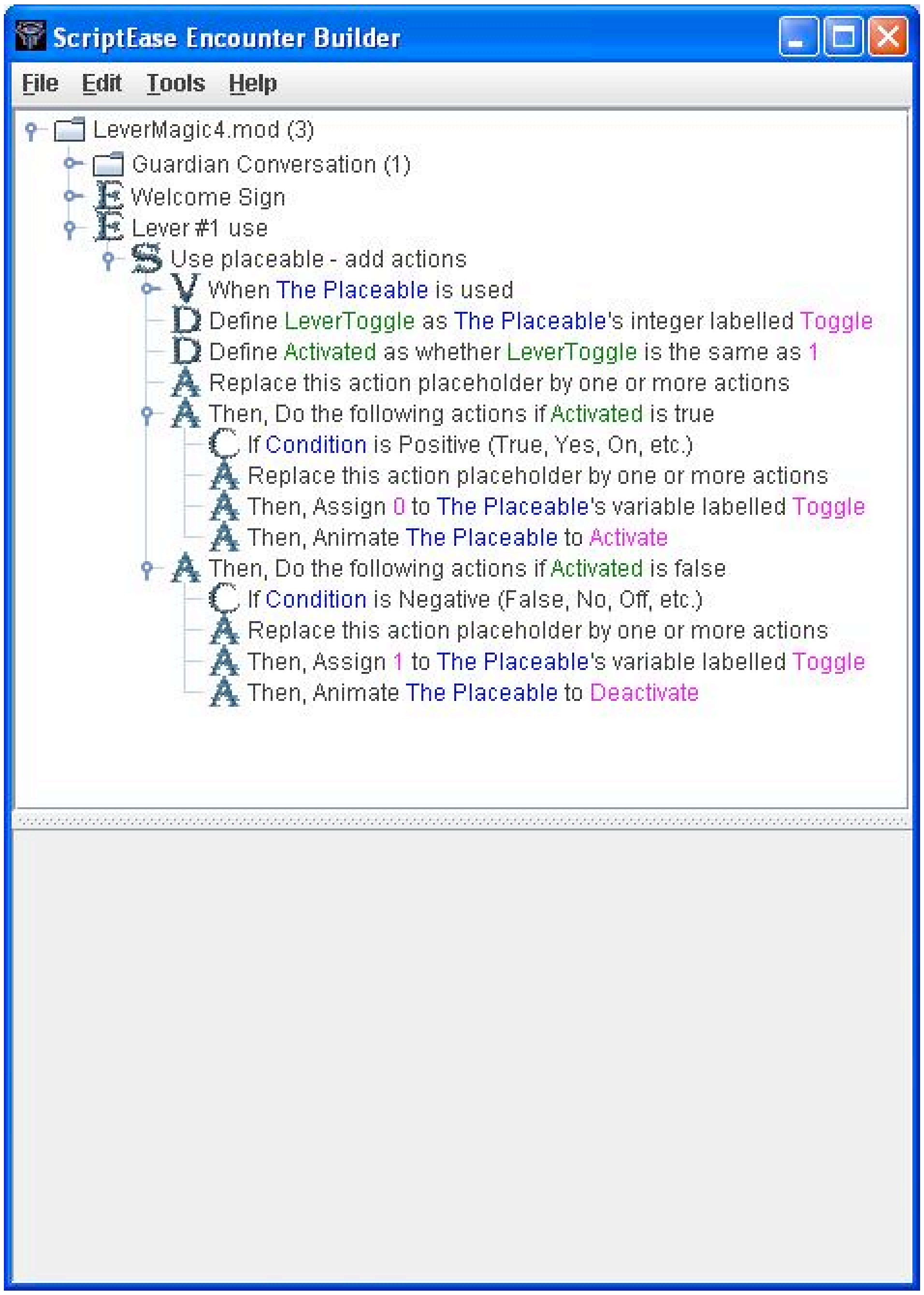
19) Under the “Label” tab-->Constant-->put “Toggle” in the textbox. Leave the value as “0”.

20) To animate the lever, simply Add an Action-->Action Atom-->Animations-->Animate a placeable object.

21) Under the “Actor” tab-->Select Placeable-->The Placeable and under the “Animation” tab-->Constant-->Activate.

22) Now that you've finished scripting the lever's actions when the “Activated” binary is true (when the lever is already in the “on” position), we need to do similar things for when “Activated” is false. We'll assign “1” to The Placeable's variable labeled “Toggle”, and we'll “Animate The Placeable” to “Deactivate”.

23) Your module should now look like the screenshot below. “Save and Compile” your module and make sure that the lever changes position each time you “use” it. Now, let's make the other levers work.



```
LeverMagic4.mod (3)
    Guardian Conversation (1)
    Welcome Sign
    Lever #1 use
        Use placeable - add actions
            When The Placeable is used
            Define LeverToggle as The Placeable's integer labelled Toggle
            Define Activated as whether LeverToggle is the same as 1
            Replace this action placeholder by one or more actions
            Then, Do the following actions if Activated is true
                If Condition is Positive (True, Yes, On, etc.)
                Replace this action placeholder by one or more actions
                Then, Assign 0 to The Placeable's variable labelled Toggle
                Then, Animate The Placeable to Activate
            Then, Do the following actions if Activated is false
                If Condition is Negative (False, No, Off, etc.)
                Replace this action placeholder by one or more actions
                Then, Assign 1 to The Placeable's variable labelled Toggle
                Then, Animate The Placeable to Deactivate
```

Getting the other levers going:

24) We still haven't got Levers 2 and 3 doing anything, but now that Lever #1 is properly configured, it is quite simple to get the others going as well. First, open your module in SE, right-click on the “Lever #1 use”, and choose “Copy” from the menu.

25) Right-click on the “LeverMagic4.mod” folder, and choose “Paste”. Repeat this to paste a second copy of “Lever #1 use”.

26) Select the first copy, under the “Description” tab-->Instance Name-->change it to “Lever #2 use”, and under “The Placeable” tab-->Module Blueprint-->Pick-->Lever #2. Repeat these steps for the second copy, changing the label and blueprint to reflect the use of Lever #3.

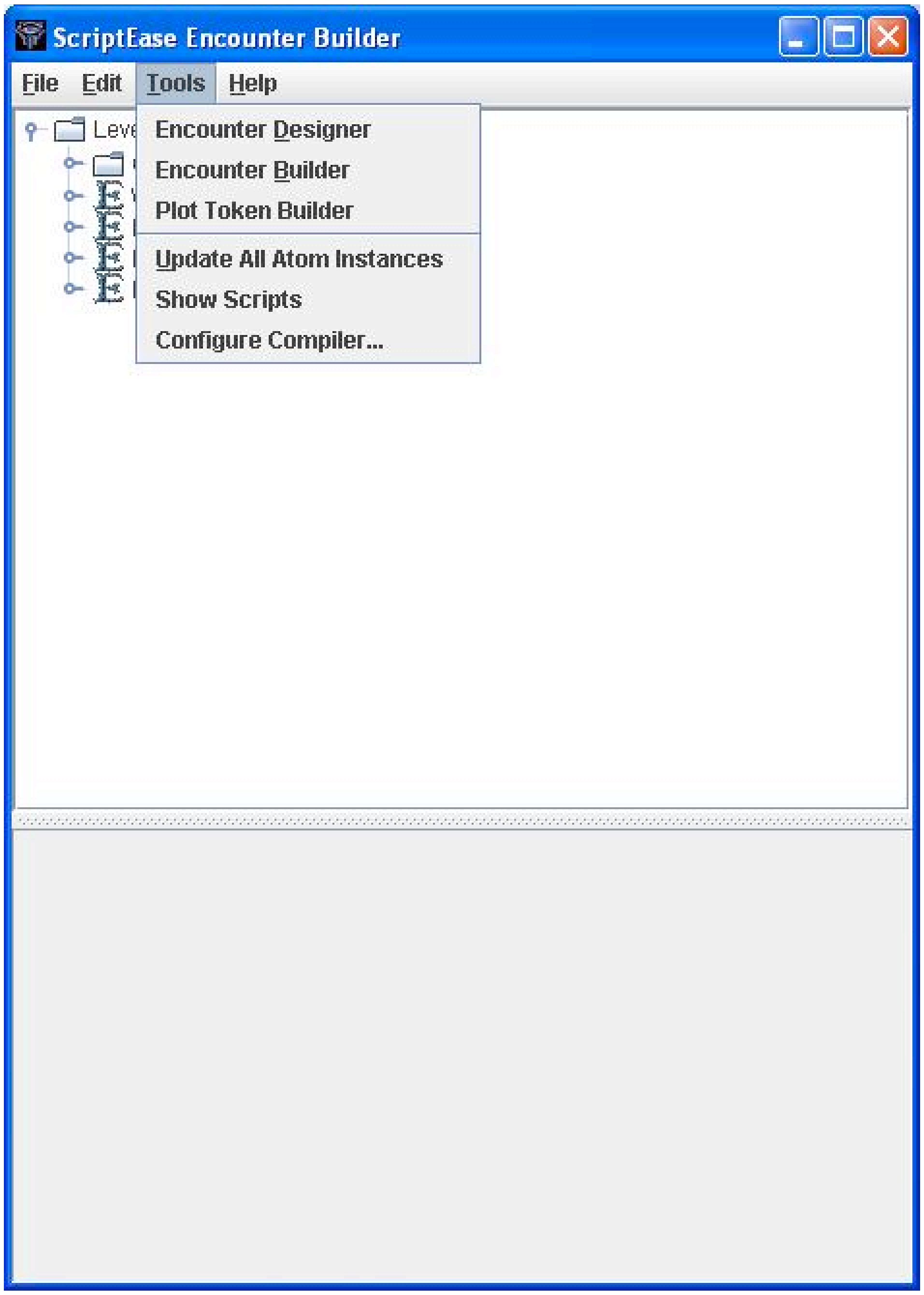
27) Now save and compile your module, and try out your newly animated levers. “So simple?” you ask? Yes. By constructing the first lever properly (by using variable names like “The Placeable”), we made it easy to reuse scripts. This is the “reason” I mentioned way back in step 8.

Plot Tokens:

Making tokens:

28) Plot tokens are sort of revisiting the notion of “state” that we discussed when animating the levers. We're going to use them so that the scripts know if the player has pulled the levers in the correct order. If you've forgotten, the order that the Guardian has prescribed is 1, 2, then 3. Let's create some plot tokens to help us enforce this order.

29) Open your module in SE (again), and choose “Plot Token Builder” from under the “Tools” menu at the top of the screen. The location of the “Tool” menu is illustrated in the screenshot below.



30) In the "Plot Token Builder" window that opens, right-click in the big white box that's in the center of the window. Choose "New Plot Token" from the menu.

31) Your new plot token will be automatically selected (indicated by a light blue highlighting of your plot token). In the "Token Name" box, put in "lever1token".

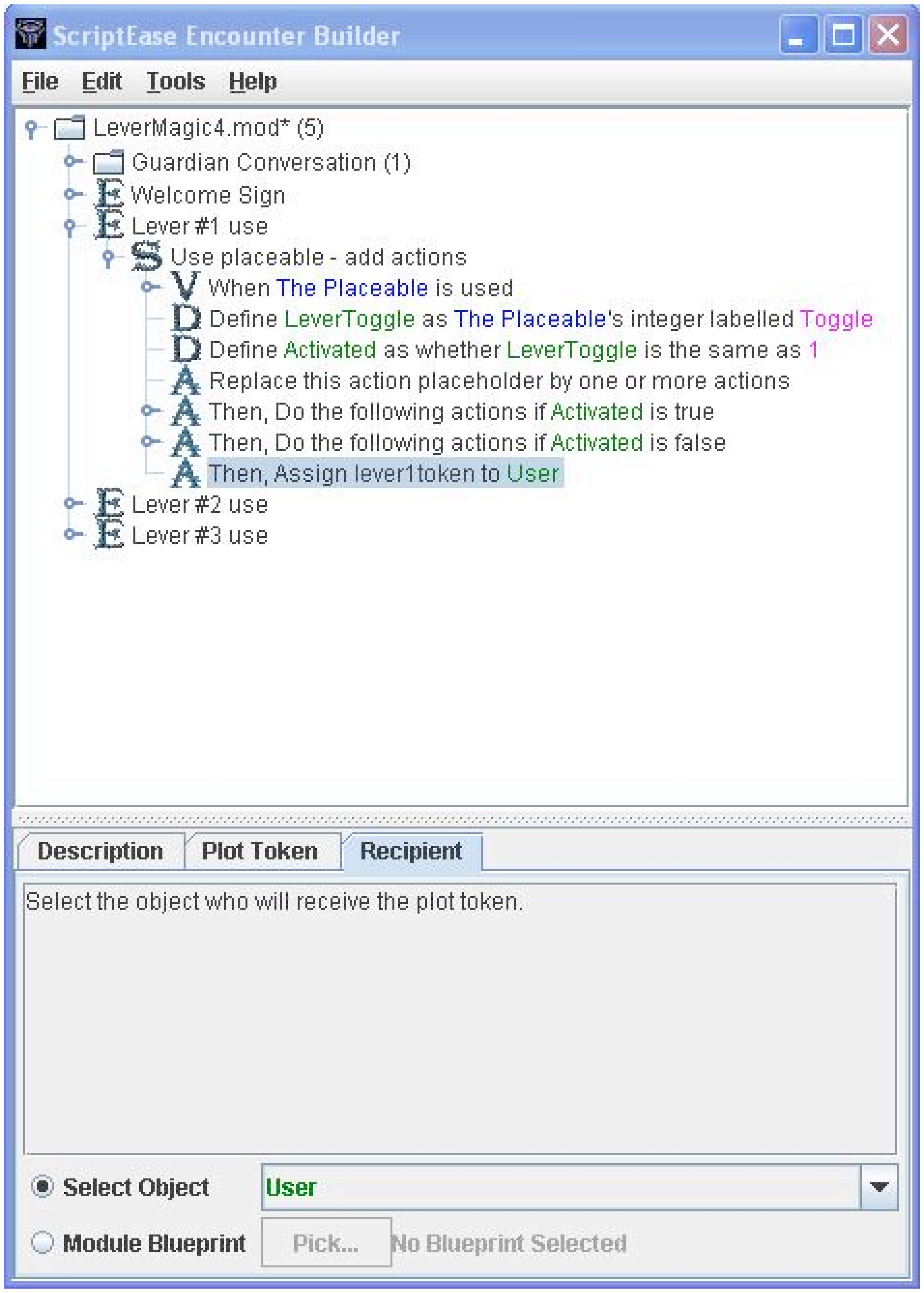
32) Repeat that steps 30 and 31 to create "ever2token".

33) Left-click the red 'X' in the top, right corner of the “Plot Token Builder” window to close it.

Getting tokens:

34) Expand the “Lever #1 use” Encounter, right-click the “Use placeable...” Situation, and Add an Action-->Action Atom-->Plot-->Assign a plot token to an object

35) In the “Action” you've just created, under the “Plot Token” tab-->Constant-->lever1token and under the “Recipient” tab-->Select Object-->User. Your module should now look like the screenshot below.


ScriptEase Encounter Builder

File Edit Tools Help

```
LeverMagic4.mod* (5)
    Guardian Conversation (1)
    Welcome Sign
    Lever #1 use
        Use placeable - add actions
            When The Placeable is used
            Define LeverToggle as The Placeable's integer labelled Toggle
            Define Activated as whether LeverToggle is the same as 1
            Replace this action placeholder by one or more actions
            Then, Do the following actions if Activated is true
            Then, Do the following actions if Activated is false
            Then, Assign lever1token to User
    Lever #2 use
    Lever #3 use
```

Description | Plot Token | Recipient

Select the object who will receive the plot token.

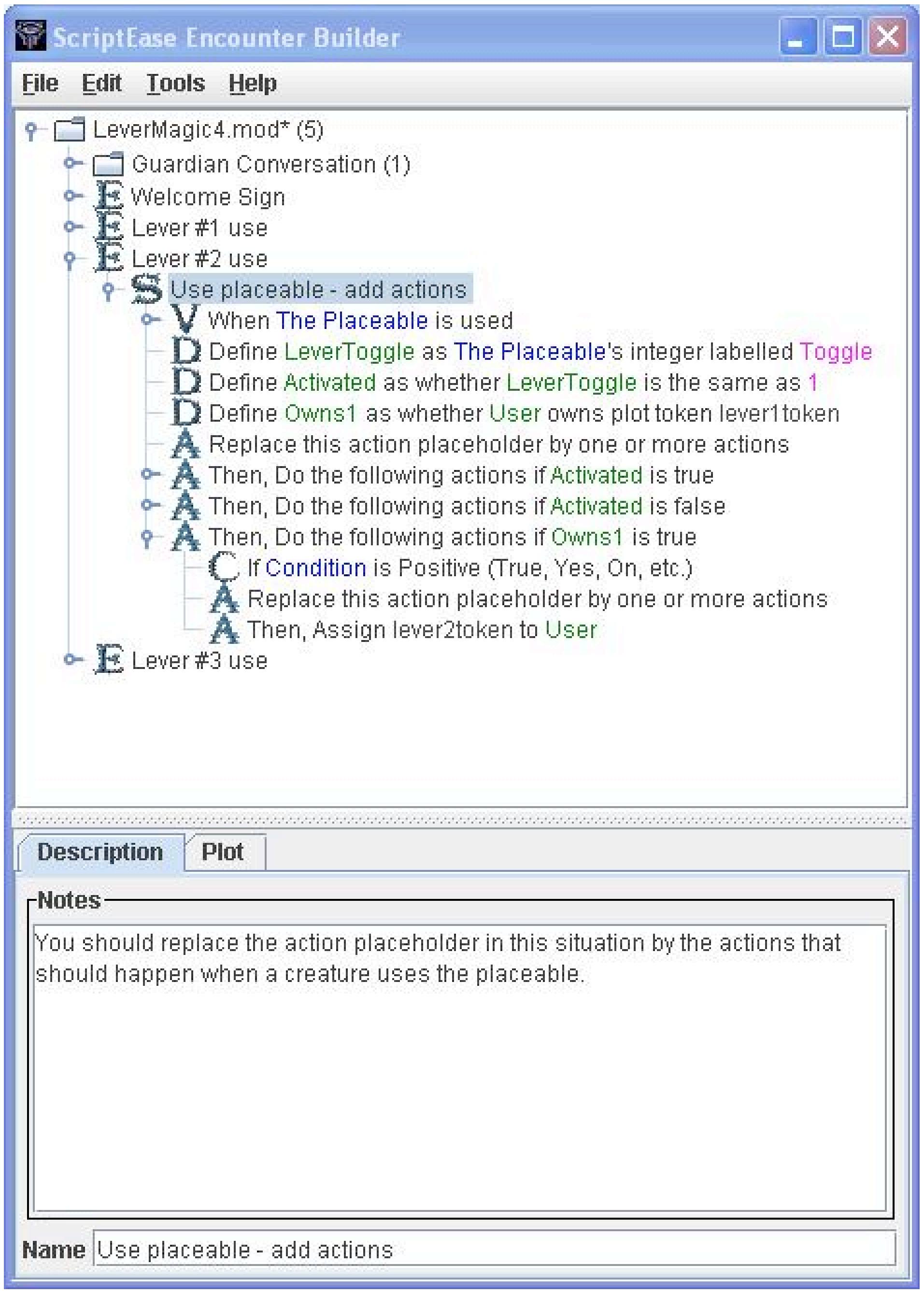
(●) Select Object: User

( ) Module Blueprint: Pick... No Blueprint Selected

36) Expand the "Lever #2 use" Encounter, right-click the "Use placeable..." Situation, and Add a Definition-->Binary-->Plot-->Whether an object owns a plot token. Under the "Description" tab-->Label-->change to Owns1, under the Owner tab-->Select Object-->User, and under "The Token" tab-->Constant-->lever1token.

37) Right-click the "Use placeable..." Situation and Add an Action-->Conditional Action-->Do...if true. Under the "Condition" tab-->Select Binary-->Owns1.

38) Right-click on the “...if Owns1 is true” Action and Add an Action-->Action Atom-->Plot-->Assign a plot token to an object. Under the “Plot Token” tab-->Constant-->lever2token and under the “Recipient” tab-->Select Object-->User. Your module should look like this screenshot:



LeverMagic4.mod* (5)
Guardian Conversation (1)
Welcome Sign
Lever #1 use
Lever #2 use
Use placeable - add actions
When The Placeable is used
Define LeverToggle as The Placeable's integer labelled Toggle
Define Activated as whether LeverToggle is the same as 1
Define Owns1 as whether User owns plot token lever1token
Replace this action placeholder by one or more actions
Then, Do the following actions if Activated is true
Then, Do the following actions if Activated is false
Then, Do the following actions if Owns1 is true
If Condition is Positive (True, Yes, On, etc.)
Replace this action placeholder by one or more actions
Then, Assign lever2token to User
Lever #3 use

Notes

You should replace the action placeholder in this situation by the actions that should happen when a creature uses the placeable.

Name: Use placeable - add actions

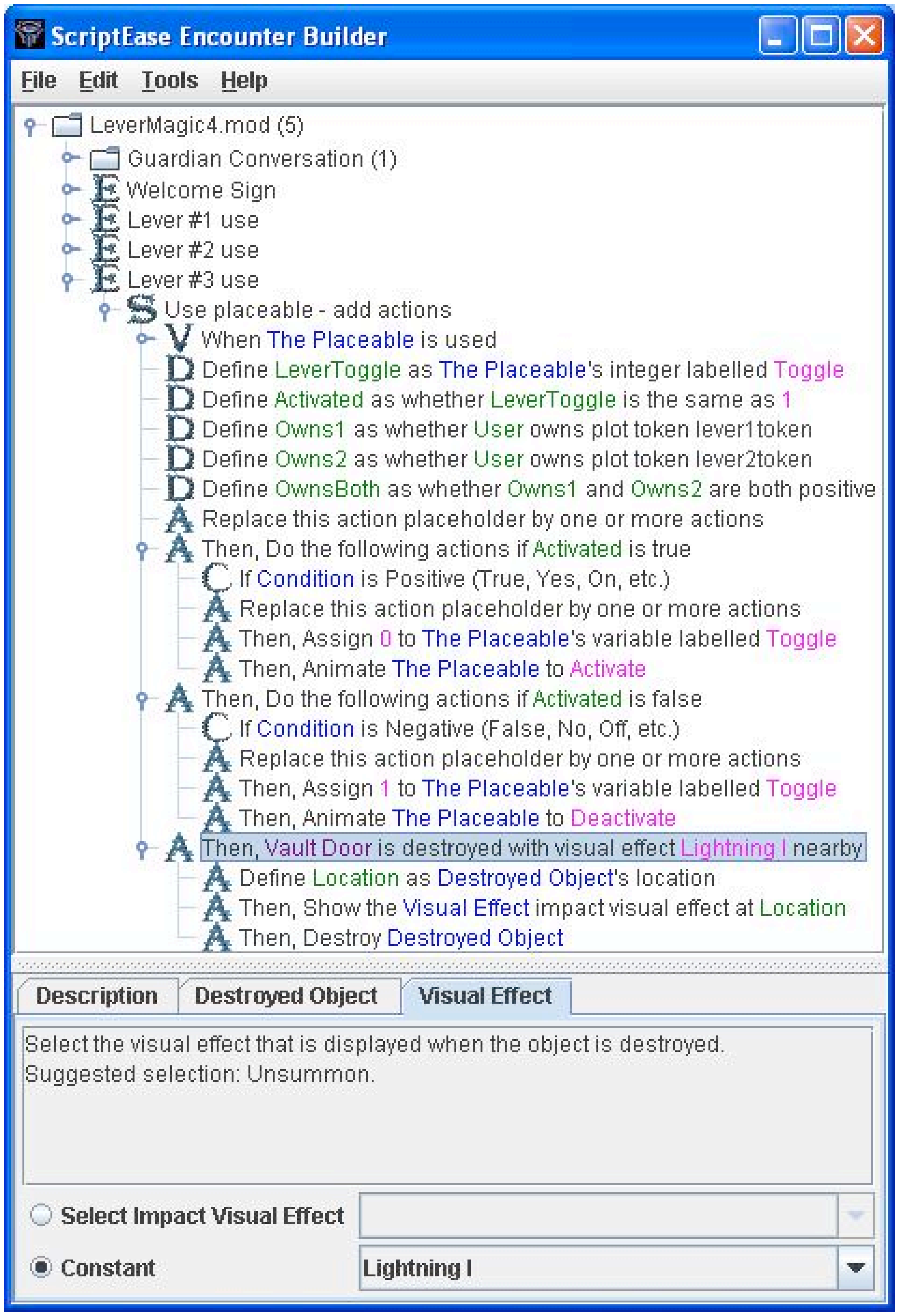
Opening doors:

39) We've made sure that the player has earned their two plot tokens, but we need something to happen when they pull that third lever. We need to open the door. This will require us to make sure that the player has both the previous tokens.

40) Repeat step 36 for the “Lever #3 use” encounter (you’ll have to re-create Owns1), and then follow a similar pattern to create a “Binary” for the “lever2token”, calling it “Owns2”.

41) Right-click the “Use placeable...” Situation and Add a Definition-->Binary-->Combining Binary Values-->Whether both of two binary values are positive. Under the “Description” tab-->Label-->change it to “OwnsBoth”, under the “First Value” tab-->Select Binary-->Owns1, and under the “Second Value” tab-->Select Binary-->Owns2.

42) Right-click the “Use placeable...” Situation and Add an Action-->Action Encounter-->Destroy an object with an effect near the object. Under the “Destroyed Object” tab-->Module Blueprint-->Pick-->Vault Door and under the “Visual Effect” tab-->Constant-->Lightning I. Your module should look like this screenshot.



ScriptEase Encounter Builder

File Edit Tools Help

```
LeverMagic4.mod (5)
  Guardian Conversation (1)
  Welcome Sign
  Lever #1 use
  Lever #2 use
  Lever #3 use
    Use placeable - add actions
      When The Placeable is used
      Define LeverToggle as The Placeable's integer labelled Toggle
      Define Activated as whether LeverToggle is the same as 1
      Define Owns1 as whether User owns plot token lever1token
      Define Owns2 as whether User owns plot token lever2token
      Define OwnsBoth as whether Owns1 and Owns2 are both positive
      Replace this action placeholder by one or more actions
      Then, Do the following actions if Activated is true
        If Condition is Positive (True, Yes, On, etc.)
        Replace this action placeholder by one or more actions
        Then, Assign 0 to The Placeable's variable labelled Toggle
        Then, Animate The Placeable to Activate
      Then, Do the following actions if Activated is false
        If Condition is Negative (False, No, Off, etc.)
        Replace this action placeholder by one or more actions
        Then, Assign 1 to The Placeable's variable labelled Toggle
        Then, Animate The Placeable to Deactivate
      Then, Vault Door is destroyed with visual effect Lightning I nearby
        Define Location as Destroyed Object's location
        Then, Show the Visual Effect impact visual effect at Location
        Then, Destroy Destroyed Object
```

Description | Destroyed Object | Visual Effect

Select the visual effect that is displayed when the object is destroyed.
Suggested selection: Unsummon.

○ Select Impact Visual Effect

● Constant: Lightning I

43) "Save and Compile" your module and check out your handiwork.

Treasure:

44) Now we're at the real reason why you would create a module in the first place: to make the best treasure possible and give it to your characters. Since the creation of treasure has already been covered (way back in the original ScriptEase tutorial), I'm not

going to repeat it here. Use your imagination. Do whatever you like but make sure to place it INSIDE the vault.

That’s all folks. If you're cagey, you may have noticed that there are a few holes in our module: you don't HAVE to talk to the guardian to get into the vault, pulling all the levers in a random order will eventually open the door, etc. These are left for you to tackle on your own as you continue to explore the world of NWN, AT, and SE.