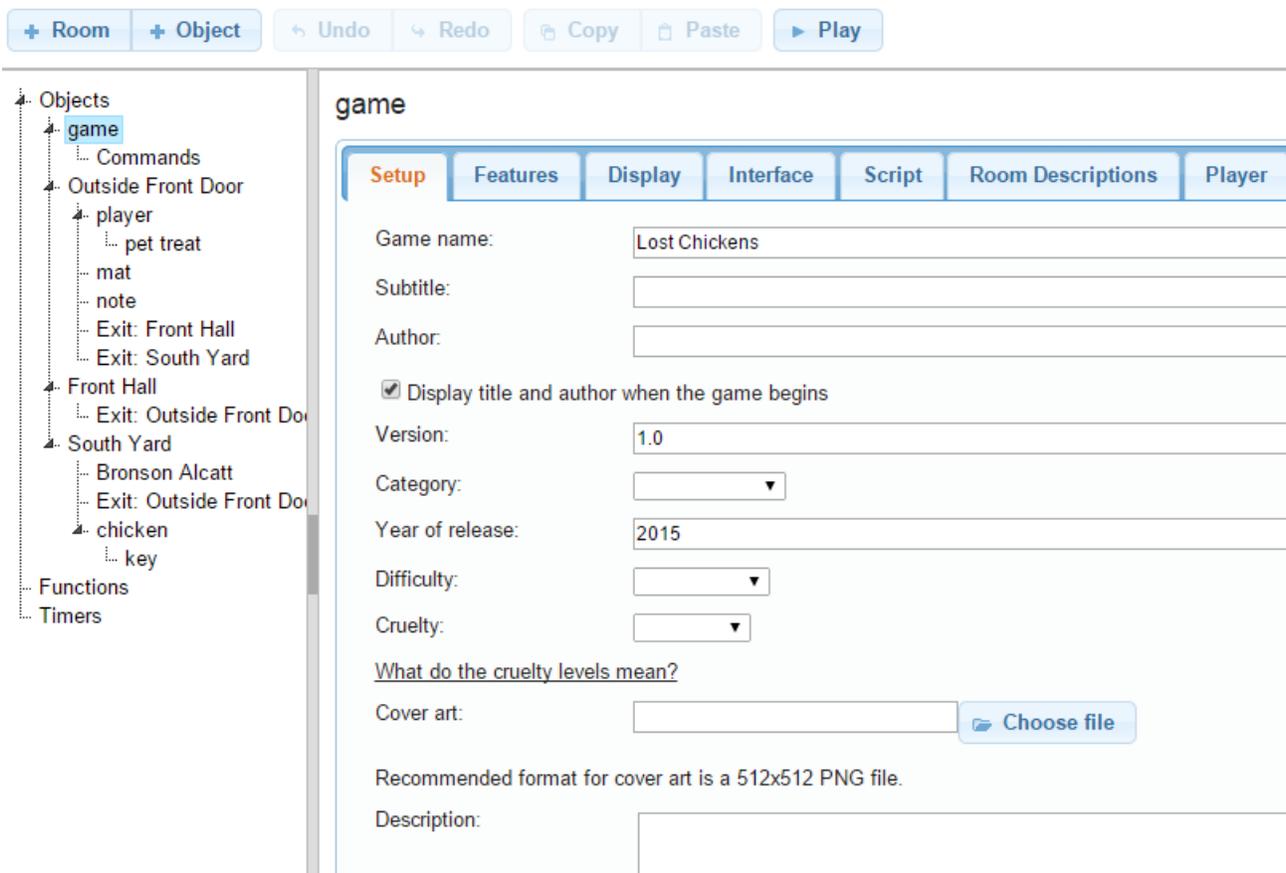


Set Up Your Story

1. Log in to your Quest account and click on the “Create” button at the top of your screen.
2. Click on “Create a New Game.”
3. In the “Game name:” box, type in the title of your story. Your “Game Type” is “Text Adventure.”
4. Click on “Create.”
5. Click on “Start editing!”
6. Wait around for a minute or two. Like many Web applications, Quest can be a bit slow to execute.
7. Quest will open with a screen that looks like this:



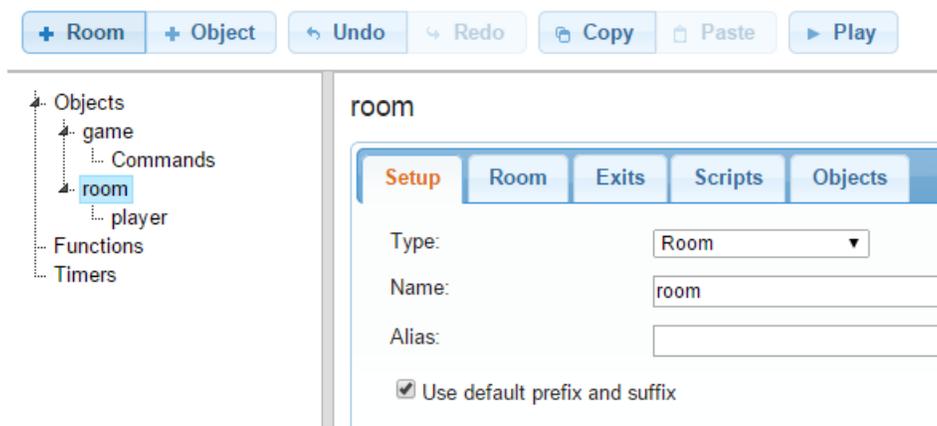
The screenshot shows the Quest game editor interface. At the top, there are buttons for '+ Room', '+ Object', 'Undo', 'Redo', 'Copy', 'Paste', and 'Play'. Below this is a tree view on the left labeled 'Objects' with a 'game' folder expanded, showing sub-items like 'Commands', 'Outside Front Door', 'player', 'pet treat', 'mat', 'note', 'Exit: Front Hall', 'Exit: South Yard', 'Front Hall', 'Exit: Outside Front Door', 'South Yard', 'Bronson Alcott', 'Exit: Outside Front Door', 'chicken', and 'key'. The main area is titled 'game' and has tabs for 'Setup', 'Features', 'Display', 'Interface', 'Script', 'Room Descriptions', and 'Player'. The 'Setup' tab is active and contains the following fields:

- Game name:
- Subtitle:
- Author:
- Display title and author when the game begins
- Version:
- Category:
- Year of release:
- Difficulty:
- Cruelty:
- [What do the cruelty levels mean?](#)
- Cover art:
- Recommended format for cover art is a 512x512 PNG file.
- Description:

8. Notice that the word “game” is highlighted on the diagram that appears on the left side of the screen. That diagram is important, and we'll have to monitor it carefully as we go along.
9. Fill in the information that's asked for on the “Setup” tab. Feel free to examine and even experiment with the other tabs if you wish, with the exception of the “Player” tab. It's best to leave that one as it is, in order to head off possible buggy behavior.

Enhance Your First Room

1. You may have noticed, in the chart on the left of the screen, that Quest has already created a room for you and placed the player/character in that room.
2. Click on the word “room” in the chart at the left. Do not click on the button labeled “+ Room” at the top of the screen. We'll use that one later.
3. A screen that looks like this will open:



4. In the screen that opens, fill in the name of the opening room, in our case “Outside Front Door.”
5. As an “alias,” use something like “outside your front door, which lies to the west.”
6. Click on the “Room” tab and fill in a description, such as “This is a grassy area outside your front door, which lies to the west.”
7. We'll use the other tabs later. For now, click on the “Save” icon at the upper right corner of the screen, unless Quest has already greyed it out, indicating that the story is already saved.
8. In addition to saving your story, you'll want to download it occasionally as your project gets larger. Downloaded versions of you story are excellent insurance, in case something goes wrong with your on-line copy.
9. The downloaded stories will require the full Quest program, a Windows-only tool, for editing them. However, in the case of a complex story, most users will be better off with the Windows program anyway, as it's faster, more complete, and more reliable.

Add Two More Rooms

1. Click on the word “game” in the chart at the left

of the screen. Make sure that the word “game” is highlighted, as we want our additional rooms to be inside the game, but not inside anything else.

2. Click on the “+ Room” button at the top of the screen.
3. You'll see a familiar room-edition screen. Fill in the “Name” as “South Yard” and the “Alias” as something like “a lovely yard, south of your home.”
4. Click on the “Room” tab and fill in the description text, “This is the lovely South Yard of your home.”
5. Click on the “Exits” tab. Since we want the player to go north from here to Outside Front Door, click on the circle beside the word “North.” When you've done so, you'll see a dropdown menu labeled “Create an exit to:” Use this dropdown menu to select “Outside Front Door.”
6. We want this exit to work in both directions, so check the box that is labeled “Also create exit in other direction.”
7. Once again, click on the word “game” in the chart that appears at the left of the screen. Click on the “+ Room” button at the top of the screen.
8. This time, we'll name the room “Front Hall,” and we'll give it the alias “cozy front hall of your home.”
9. We'll not use the “default prefix” for this room.

Instead we'll use the prefix “the.”

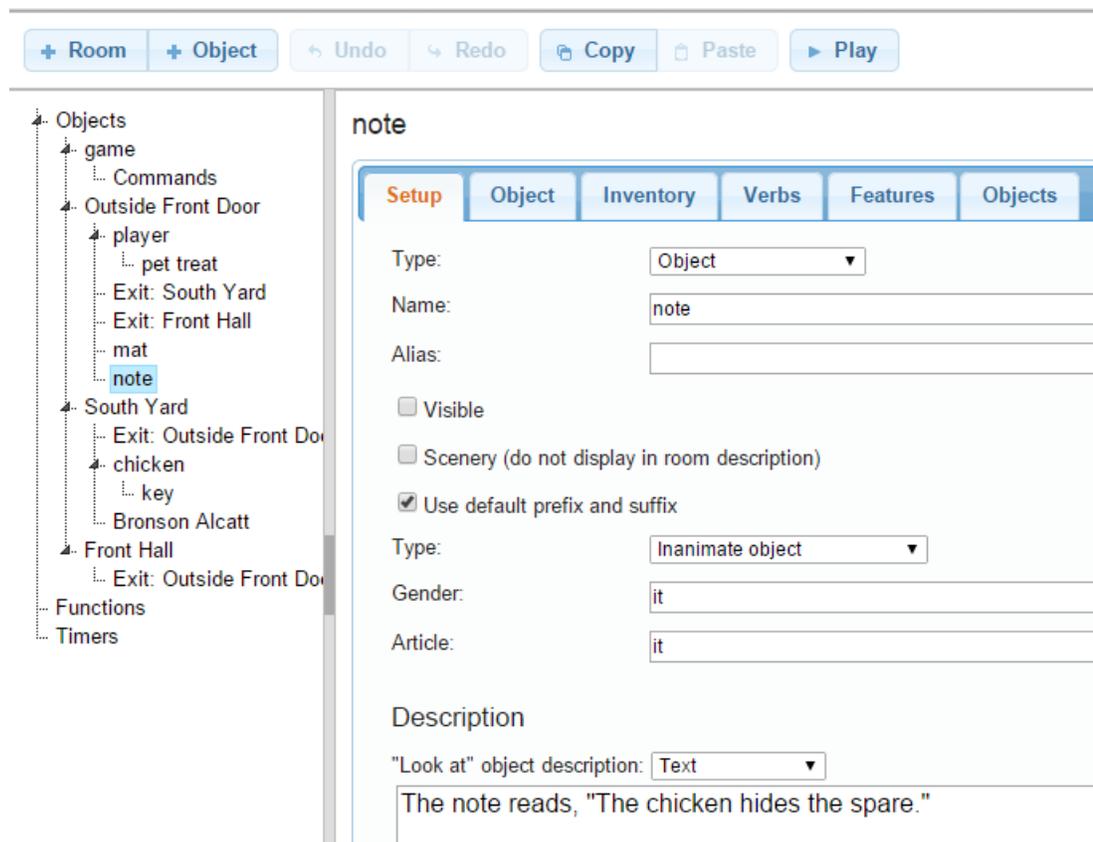
10. We won't use the “Room” tab here, but we will click on the “Exits” tab.
11. With the “Exits” tab open, create a two-way exit that leads east to “Outside Front Door.” Later, we'll create a restriction on when the player can go west from Outside Front Door to Front Hall.

Try Out Your Story, So Far

1. Click on the “Save” button in the upper right part of the screen, unless the button is already greyed out.
2. Click on the “Play” button at the top of the screen. After a few seconds, you should see you story running, with the player in the room called “Outside Front Door.”
3. You should be able to move around through the three rooms you've created.
4. Occasionally, the Quest interpreter seems to generate odd error messages, in response to common IF commands. Often, simply trying your command a second time will solve the problem.

Create Two Objects, the Doormat and the Note

1. New writers of interactive fiction often ask about how a writer can cause a concealed object to appear, the way the note appears in “Lost Chicken,” when the player/character takes the doormat. Here's one of the several possible ways to get this effect.
2. First, let's create the note.
3. Click on “Outside Front Door” on the chart that appears at the left of your screen. You will see the familiar screen which you used to provide information about the “Outside Front Door” room.
4. On this screen, click on the “Objects” tab. A screen will open, showing that one object, the player, is already in “Outside Front Door.”
5. In the “Objects” tab, click on “Add.” A screen will open, allowing you to provide information about your new object.
6. Give your new object the name “note.” Make its description, “The note reads, 'The chicken hides the spare.'”
7. Click on the “Inventory” tab, and indicate that the new object can be taken.
8. Make sure that the box beside “Visible” is not checked. The note, in its initial form, will not be visible to the player, but we'll make it visible soon.



9. Once again, click on “Outside Front Door” on the chart at the left of the screen. Then click on the “Objects” tab.
10. Click on Add.
11. This time, call your object “mat,” and make its alias “doormat.” Its description should be something like “an ordinary doormat.”
12. Click on the “Object” tab and then click on the “+ Add” button that appears beside “Other Names.” Add the new name “mat.”
13. Click on the “Inventory” tab and indicate that the mat can be taken.
14. Click on the “Save” button if it's not already greyed out.

Create Your First Script

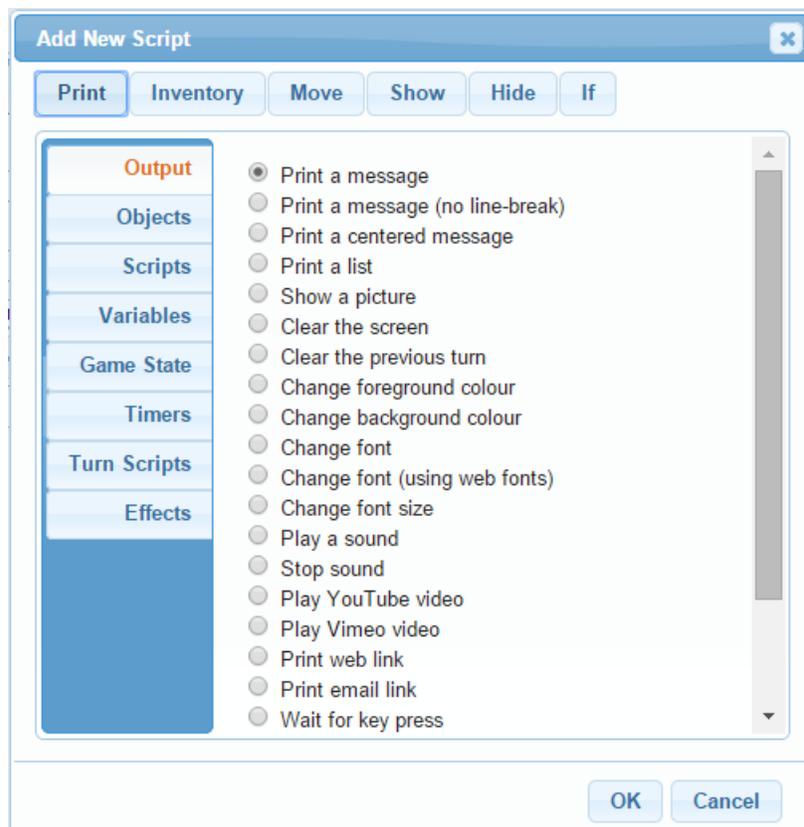
1. Some Background Concepts

1. Scripts are central to Quest, in pretty much the same way that tasks are central to Adrift. In Inform 7, which we'll consider later in this book, rules serve more or less the same purpose as scripts.
2. Scripts can be quite complex, but they are essential.
3. Scripts allow an author to change the default behavior of the game world. For example, in our "Lost Chicken" story, we can change what happens when the player/character takes the mat, causing the note to appear.
4. As we build a script, it we will often add other scripts to it. Thus, it is extremely common for an author to start creating a script and then to add other scripts to it.

2. Our First Script – the Goal

1. In order to make the note appear when we want it to, we'll build a script. These are the characteristics of that script.
 1. It takes place only after the player/character takes the mat when the note has not yet been revealed. We don't want the note to mysteriously reappear every time the player/character drops mat and picks it up again.

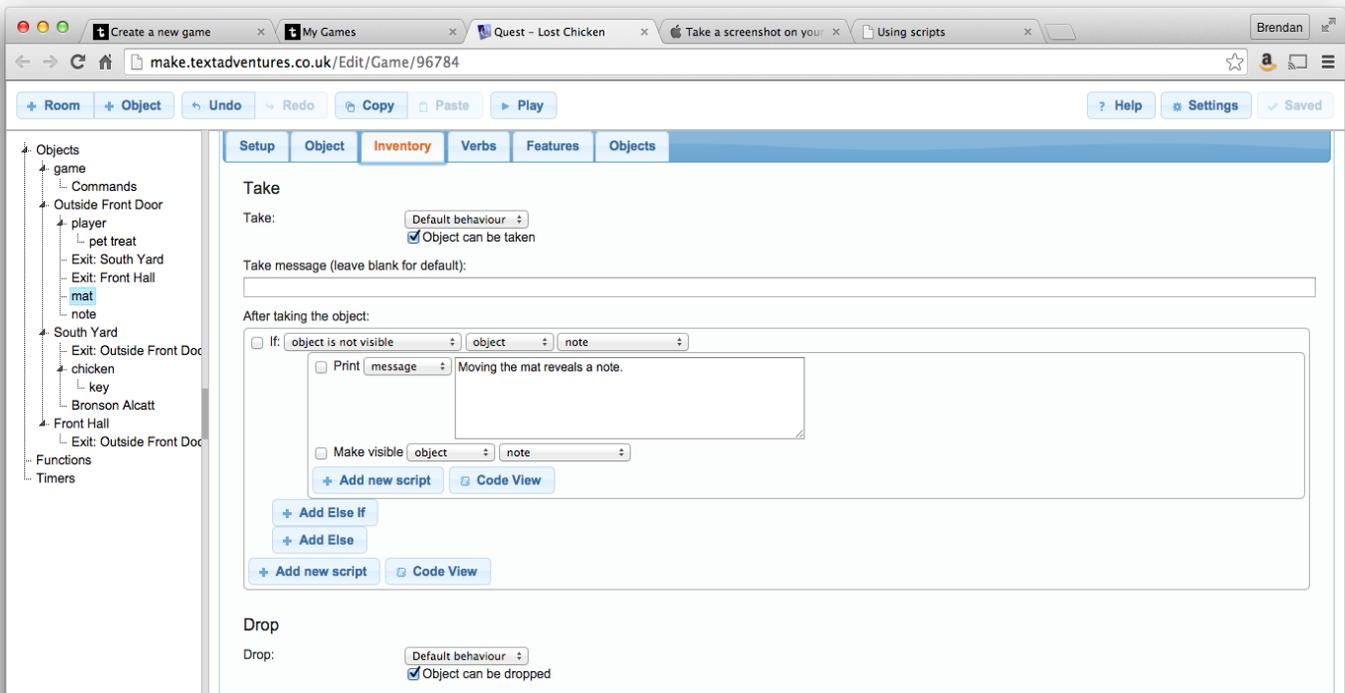
2. It causes the note to appear.
3. It tells the reader what the previous elements of the script caused to happen.
2. We'll build our script, using the "Inventory" tab for the object called "mat."
3. Our First Script – the Building Process
 1. On the chart that appears at the left of your screen, click on "mat."
 2. Click on the "Inventory" tab.
 3. Under "After taking the object," click on the button labeled "+ Add new script." A popup window that looks very much like this will open:



4. Click on the word “If,” which is in the upper right part of the popup window.
5. The word “IF” will appear, followed by a series of dropdown menus.
6. Using the dropdown menus, create an “if” clause that reads, “IF: object is not visible object note.”
7. By now, you've probably noticed that the screen you're looking at has become pretty complicated. For one thing, it contains at least three identically-labeled buttons that read “+ Add new script.”
8. One of these “+ Add new script” buttons is directly under “If: object is not visible object note.” And this particular button is indented a bit.
9. The indenting is intended to show that whatever command (AKA “script”) you add using the indented button will be carried out only if the note is not visible.
10. This sort of indenting is of paramount importance in creating good scripts. Such scripts often require nesting groups of commands within other groups of commands, and the indenting is very helpful in keeping all the nests sorted out.
11. Click on the indented button that's labeled “+ Add new script.” A popup window will open.”
12. The defaults that are already selected in the popup window will probably indicate that you want to print a message. As a matter of fact, that's a good idea here, so just click on “OK.” If the defaults do not have “Print” and “Print a message”

already selected, select them before you click “OK.”

13. Indicate that the message you want to print is something like “Moving the doormat reveals a note.”
14. Notice that there's now a “+ Add new script” button directly under the word “Print.” Click on that “+ Add new script” button.
15. This time, on the popup window, select the “Objects” button and choose “Make object visible.” Click on “OK.”
16. Now using the dropdown menus, assemble this command, “Make visible object note.”
17. Your screen should now look like this:



18. Click on “Save” if it's not already greyed out. Then try out your story so far, using the “Play” button.

Create Your First Container

1. IF authoring systems usually have some built-in object types. One of these is the container, a type of object that can hold other objects. In “Lost Chicken,” the plaster chicken is a container that has a key in it.
2. To create the plaster chicken, begin by clicking on “South Yard” in the chart on the left side of the screen. Then click on the “Objects” tab.
3. Click on “+ Add.”
4. A familiar screen for creating an object will

- appear, with the “Setup” tab selected. Fill in the name “chicken” and the alias “plaster chicken.” For a description, type something like “An old plaster chicken, used as a container.”
5. With the “Object” tab depressed, add “chicken” as an “other name.”
 6. Click on the “Inventory” tab. Make sure that “Object can be taken” is not checked.
 7. Click on the “Features” tab and select “Container: object is a container or surface, or can be opened and closed.”
 8. A new tab, labeled “Container,” will appear. Click on this tab, and chose, as the “Container type,” “Openable/Closable.” Then, check “Can be opened,” and “Can be closed.”
 9. Click on the “Objects” tab and then on “+ Add.” Add the object “key,” and fill in its details in our now-familiar fashion.
 10. Try out your story. It should be possible for the player/character to go south to the South Garden, open the plaster chicken, and take the key.

Create a Script That Ends Your Story

1. Our story ends when the player/character gains entry to the Front Hall. It should be impossible for the player/character to get to the Front Hall unless he or she is carrying the key.

2. We'll use a script to implement this behavior.
3. In the chart at the left of the screen, locate "Outside Front Door," but don't click on it.
4. Under "Outside Front Door," locate "Exit: Front Hall," and click on it.
5. Put a checkmark beside "Run a script (instead of moving the player automatically)."
6. As we did on our previous script, start by choosing "If" from the upper right part of the popup menu, and then build a statement, using the dropdown menus that will appear. The statement should be, "If player is not carrying object object key."
7. Once again, you'll see a fairly complex screen with a number of buttons labeled "+ Add new script." Pick out the "+ Add new script" button that is indented under your "If" clause, and click on that button.
8. Again, a popup window will open, and, as it happens, the default behavior, printing a message, is what we want. Click on "OK," and fill in the message, "You can't get through the door without the key."
9. Now, we'll add some unfamiliar features to our script. First, click on the "+ Add Else" button, which appears below your "If" clause.
10. Identify the "+ Add new script" button that is immediately under the word "Else" and slightly indented. Click on that button.

11. A familiar popup window will open. Click on “Move” at the top of the window.
12. Using dropdown menus, build the command “Move object object player to object “Front Hall.”
13. Identify the “+Add new script” button that appears immediately under the command that you just created. Click on that button.
14. Click on “OK.” Then fill in the “Print” box that appears with a message like, “Congratulations! You've finished the story.
15. Press the “+Add new script” button that appears below your “Print” statement. A familiar popup window will open.
16. Click on the “Game State” button that appears along the left side of the popup window. Then choose “Finish the game” and click on OK.
17. Save your story and try it out. You should be able to reach the end of the story.

Create Your First Non-Player Character

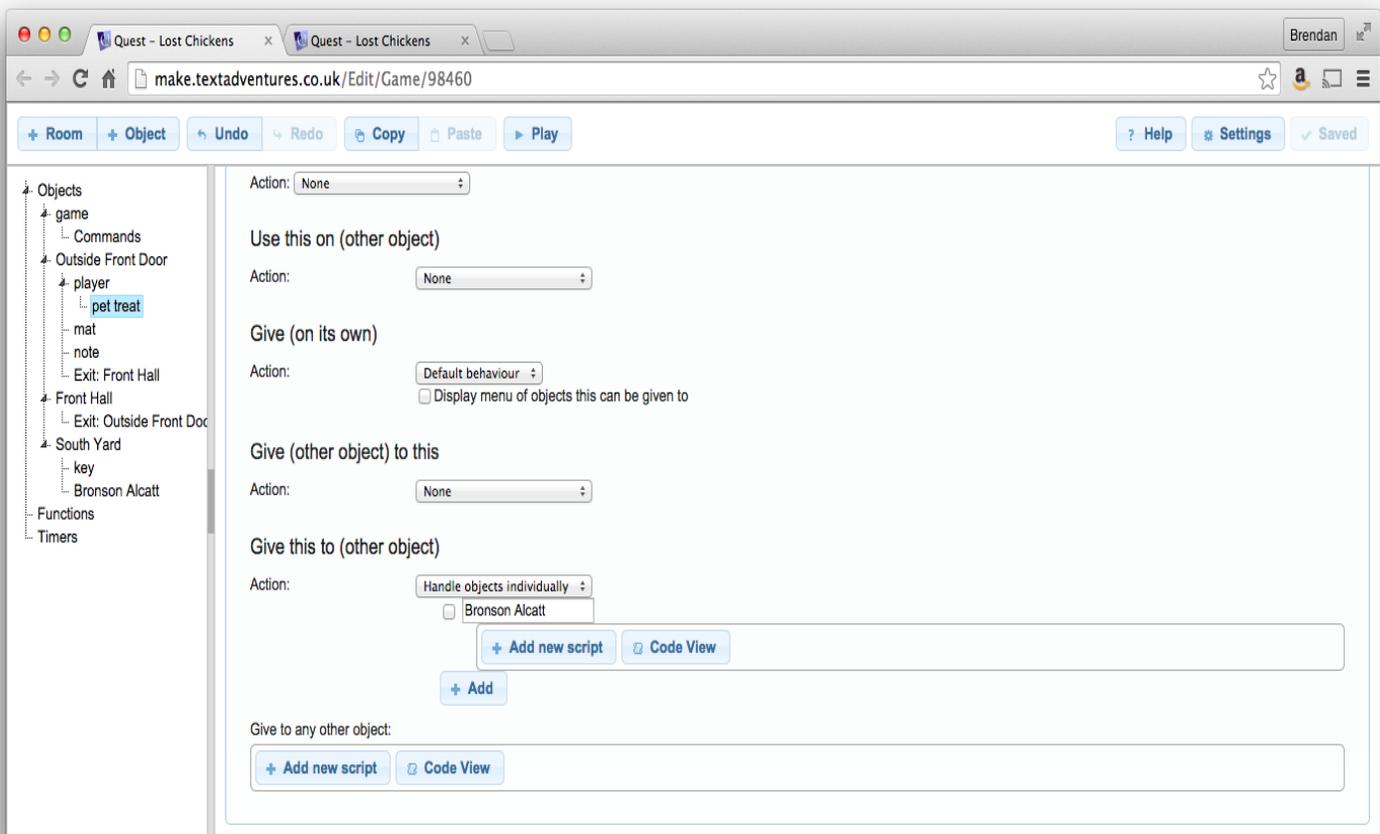
1. The only non-player character in our story is cat named Bronson Alcott. He starts out in the South Yard, so let's start out by clicking on “South Yard” on the chart at the left of the Quest screen.
2. Click on the “Objects” tab and then on the button labeled “+ Add.” Let's call our new object

- “Bronson Alcott” and use “Bronson Alcott” as his alias. Bronson's “Type” will be “Male character (named).”
3. As a “Look at” description, let's choose “Text” from the dropdown menu and type something like, “Your ancient and formidable feline.”
 4. Using the “Object” tab, let's add some names for the cat. These might be “Bronson,” “Alcott,” and “cat.”
 5. Save your progress, unless Quest has already saved it for you. Try out your story once more.

Create the Pet Treat

1. Quest considers the pet treat to be a bit different from the other objects we have created because we want to be able to give it to Bronson.
2. Click on the word “player” on the chart at the left of the screen. Then click on the “Objects” tab.
3. Click on “+ Add,” and give your new object the name “pet treat.” Using the dropdown menu, indicate that the “Parent” of the pet treat is the player, not the room in which the player starts out.
4. Give the “pet treat” the alias “treat.”
5. Click on the “Object” tab and add the name “pet treat.”
6. Click on the “Features” tab and check the box

- beside “Use/Give: object can be used on its own or with other objects/characters or given to other objects characters.” You'll notice that a new tab, labeled “Use/Give” will now appear.
7. Focus on the section of the screen labeled “Give this to (other object).” Beside the word “Action,” you'll see a dropdown menu. From this menu, choose “Handle object individually.”
 8. Click on “+ Add” and choose “Bronson Alcott.” Note that, slightly indented under “Bronson Alcott,” there is a familiar “+ Add new script” button.
 9. We are now going to build a script will accomplish two goals.
 1. It will tell the reader what happens when the player/character gives the treat to Bronson.
 2. It will actually transfer the treat from the player/character to Bronson.



10. Click on the “+ Add New Script” button that appears, slightly indented, under “Bronson Alcott.”
11. In the popup menu that opens, you'll notice that printing a message is already selected. As a result, simply click on “OK” at the bottom of the popup.
12. In the “Print message” box, type something like “Bronson accepts the treat and devours it greedily. He looks less grumpy now.”

13. Locate the “+ Add new script” button that appears directly below the word “Print.” Click on that button.
14. In the popup window that opens, click on “Move,” which appears on the top row of options. The popup will disappear, and you'll find that you're back on the previous page. However, you'll note that a new line has been added under your “Print” box. This line offers a series of dropdown menus.
15. Using this new line and its dropdown menus, create the command, “Move object pet treat to object Bronson Alcott.”
16. Save your work and try out your story.

Implement Bronson's Gumpy Behavior

1. We've now made it possible for the player/character to give the pet treat to Bronson, but we have not yet provided any motivation for doing so. We now need to create a way to implement Bronson's mood and to change it once he gets the treat.
2. In the various IF authoring systems, the most obvious way to implement this sort of change is to use a variable. If we were working with the Windows version of Quest, we would use a similar approach, but the Web version of Quest does not give us this option.

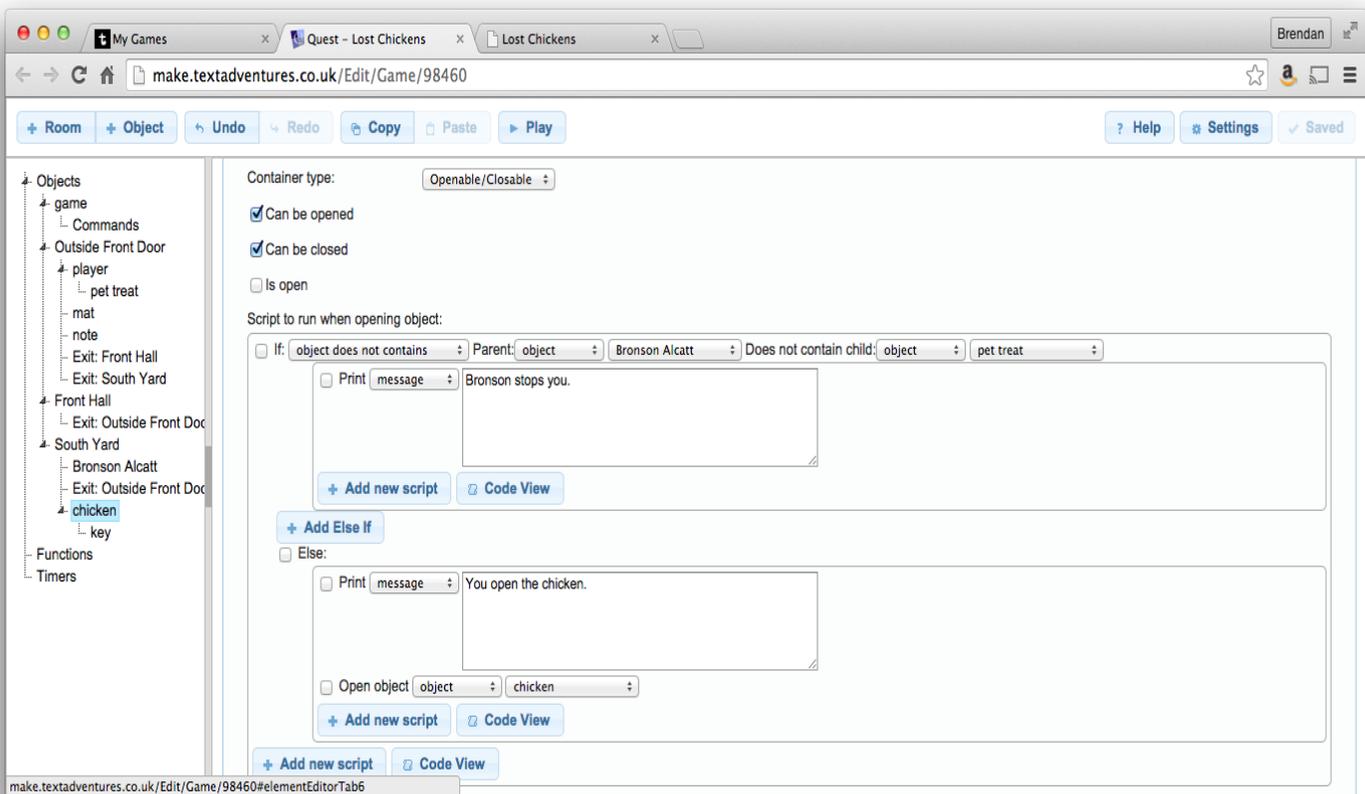
3. Instead, we'll implement Bronson's mood by keeping track of whether or not he has the pet treat.
4. Click on the word "chicken" in the chart that appears on the left of the screen. Click on the "Container" tab.
5. Under "Script to run when opening object," click on the button labeled "+ Add new script."
6. Using the dropdown menus, construct the following, not too grammatical, clause:

If: object does not contains Parent: object
Bronson Alcott Does not contain: Child object
pet treat.

7. The meaning of your odd-sounding "if" clause is, "If Bronson Alcott is not carrying the pet treat."
 1. Note that, when the player/character gives Bronson the pet treat, the reader is told that Bronson ate it.
 2. However, we never actually implemented anything that would make the pet treat go away.
 3. We did implement the transfer of the pet treat to Bronson.
 4. By default, when the player/character looks at another character, the player/character does not see what the other character is

carrying. For this reason, the reader has no way of knowing that Bronson is actually carrying the treat.

8. Locate the button labeled "+ Add new script," the one that is slightly indented under the "if" clause you just created.
9. On the popup window that opens, click on "Print" in the top left part of the popup. Then click on "OK."
10. Fill in the resulting print box with something like, "When you reach toward the chicken, Bronson tries to scratch you, preventing you from opening it. Perhaps he's grumpy because he's hungry."
11. Click on "Add Else."
12. Identify the "+ Add new script" button that appears below "Else:" and slightly indented. Click on that button.
13. In the popup window that opens, click on "Print." Fill in the print box with something like, "You open the plaster chicken."
14. Locate the "+ Add new script" button that appears directly below the word "Print." Click on that button.
15. In the popup window that opens, click on the word "Objects," which appears on the left side of the popup. Then select "Open object" and click on "OK."



16. Then using the dropdown menus that appear, construct the command, “Open object object chicken.”
17. Save your story and try it out.

Some Further Challenges

1. Try to make Bronson's description more flexible, using a script rather than plain text.
2. Implement the front door. There's a way to do this sort of implementation in the Quest documentation at http://docs.textadventures.co.uk/quest/tutorial/using_lockable_exits.html. However, you probably don't need a fully-implemented door for our story.