

## Chapter 21

# Creating Interactive Biology Tutorials for the World Wide Web Using a Simple Java Editor

*William Udell*

Center for Instructional Innovation  
University of Nebraska-Lincoln  
212 Bancroft Hall  
Lincoln, NE 68588-0384

**Reprinted From:** Udell, W. 2000. Creating interactive biology tutorials for the World Wide Web using a simple Java enabler. Pages 416-429, *in* Tested studies for laboratory teaching, Volume 21 (S. J. Karcher, Editor). Proceedings of the 21st Workshop/Conference of the Association for Biology Laboratory Education (ABLE), 509 pages.

- Copyright policy: <http://www.zoo.utoronto.ca/able/volumes/copyright.htm>

Although the laboratory exercises in ABLE proceedings volumes have been tested and due consideration has been given to safety, individuals performing these exercises must assume all responsibility for risk. The Association for Biology Laboratory Education (ABLE) disclaims any liability with regards to safety in connection with the use of the exercises in its proceedings volumes.

©2000 William K. Udell

## Contents

Introduction, A Brief History.....	417
Jamba and Java .....	417
Getting Started: Creating a Basic Java Applet with Jamba .....	417
Concept Map.....	422
Using Animation and the "To Do" List .....	425
Adding the Applet to an HTML Page .....	429

## Introduction

### A Brief History

At inception the internet was created for scientists to share data and knowledge worldwide. This data was passed through clunky gopher systems and viewed by simple terminals. Eventually the complexity of the data that needed to be shared facilitated the creation of HTML. Hypertext Markup Language (HTML) allowed simple formatting of pages that could be read by interpreters such as Mosaic from NCSA. This elegant method of sending information exploded into a revolution of communication. As the technology grew in popularity it became apparent that HTML would not be sufficient to give the sort of interaction that the users demanded. Several enhancements to HTML emerged such as JavaScript, DHTML, and Shockwave.

Simultaneously a group at Sun Microsystems, working on a project called OAK, had a revelation. They postulated that their lightweight, robust, platform independent programming language would meld with the vision of the web's future. On a hunch, Sun changed the focus of their OAK research to web development. A team member also decided that OAK wasn't artsy enough, so one very late night over a cup of, you guessed it, coffee, Java was born.

### Jamba and Java

Jamba, a commercial Java editor from Interleaf, is a set of canned Java objects that can be brought together within the comfort of a tidy user interface. With very little effort, the lay person can design Java applets with tremendous functionality. The code created by Jamba is stored in JAR archive files that are uploaded to the server. Server maintenance such as uploading Jamba files and Java class files is accomplished using a wizard included in the package.

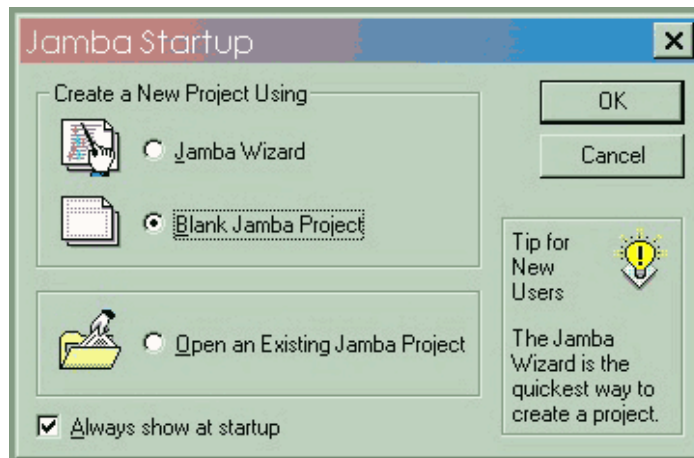
### Getting Started: Creating a Basic Java Applet with Jamba

This exercise will demonstrate some of the basics of creating Java applets with Jamba. The applet will be a drag and drop applet. Drag and drop is terminology used to describe the movement of one object to another object. In this instance, the item will "stick" if it is dragged to the appropriate location. The final product is the word CAT where the letter "A" is the dragged object. If "A" is dragged anywhere but right between "C" and "T" it will return to its original location.

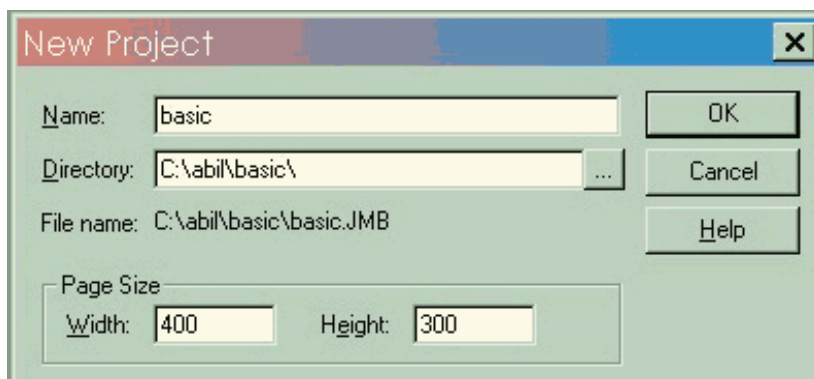
## Creating Web Tutorials

Pay particular attention to the steps involved in setting "hot spots" and "dragdrop targets." Hot spots are areas of the applet that can be used to trigger an action, and can be used as dragdrop targets, among other things.

1. Double-click the Jamba icon on the desktop.
2. When the program opens pick "Blank Jamba Project." Then click "OK."



3. Next, create the directory and file name. Call this project "basic."

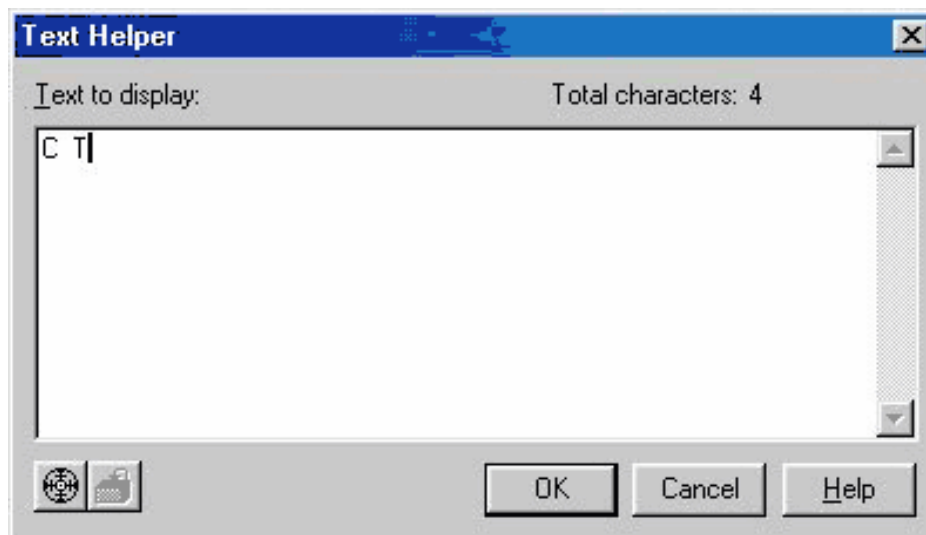


4. On the toolbar select the text icon "A"



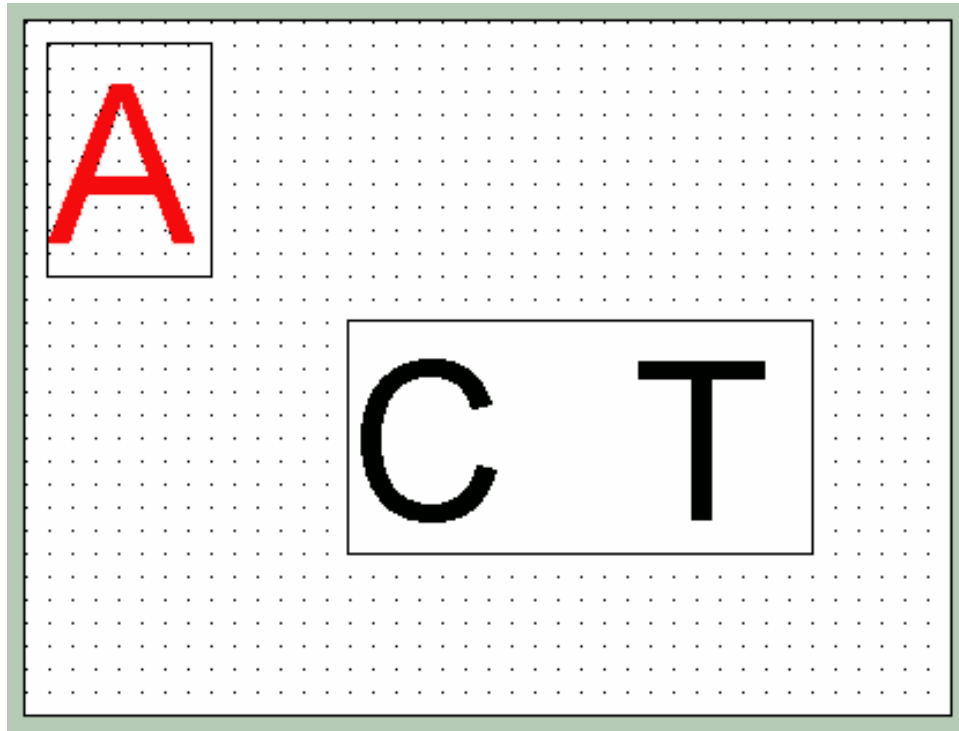
Then click somewhere on the canvas.

5. In the text help box type "C T" then click "OK."



## Creating Web Tutorials

- Repeat the last two steps this time replacing "C T" with "A." Then adjust the size of both text objects with the properties toolbar.



- Click the Hotspot icon, then click and drag a rectangle to create a hot spot between the C and T.



8. The hot spot area can now be used as a dragdrop target. Start by clicking on your text letter "A." In the properties box select CanBeDraggedTo.
9. Click on the target selector button in the Object Selector, and while still holding down the mouse button slide up and select the hot spot that was just created between the letters "C" and "T." Then click OK in the Object Selector.

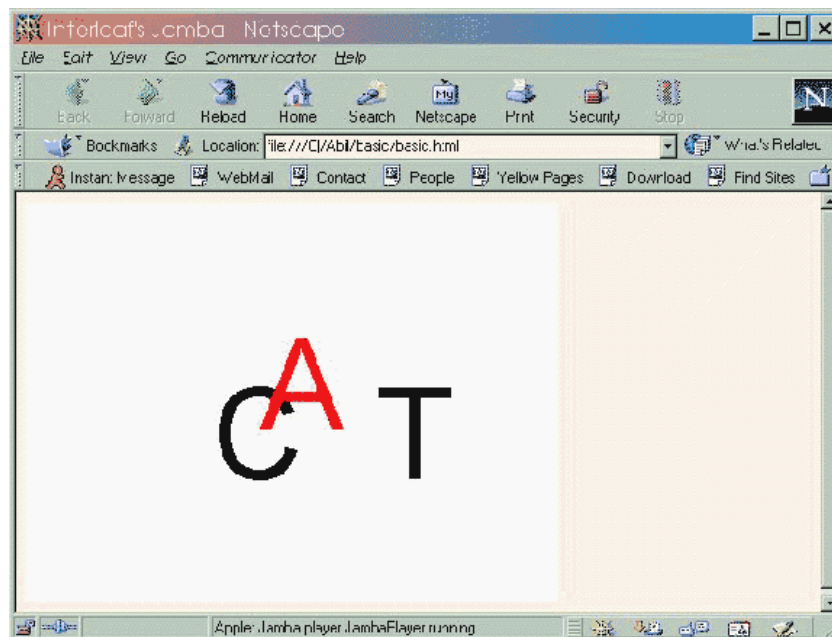


10. The next step is to try out the applet to see if it worked. Click the play button on the tool bar. This will open a Java applet viewer and load the project. The settings can also be changed to open the project in the browser of choice. For example, to change the viewer to Netscape click on "Tools" on the toolbar, then options, then run. From "run application using" select "browser." Then browse for the path to Netscape, which is usually found at:

**C:\Program Files\Netscape\Communicator\Program\netscape.exe .**

## Creating Web Tutorials

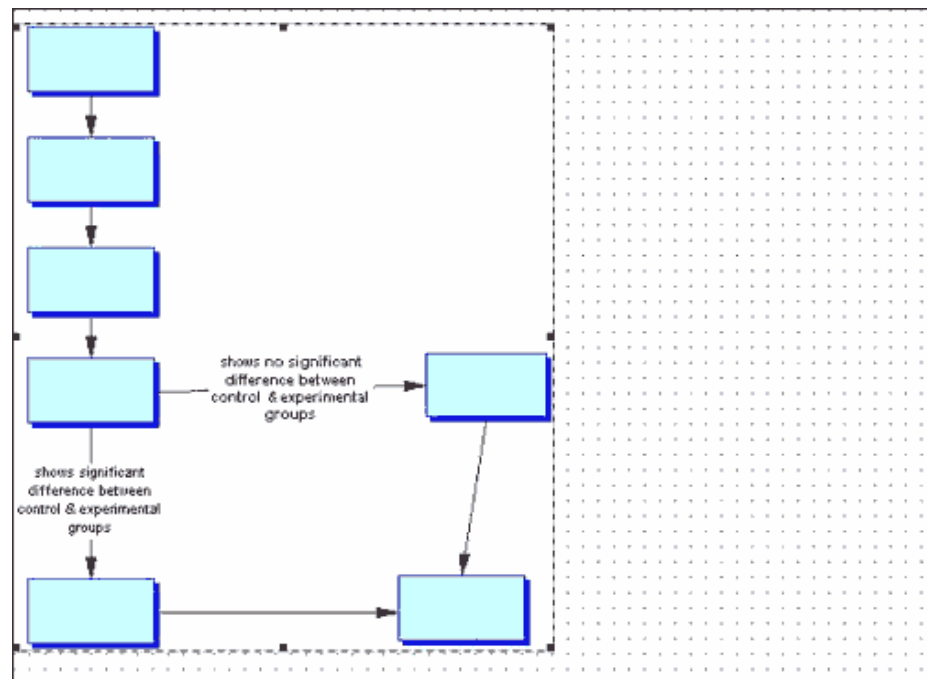
This is an example of how a final product should look when run in Netscape.



### The Concept Map

This project illustrates the use of Jamba to create interactive concept maps. The concept map used for this example was created using Inspiration™ which is a commercially available product for creating flow charts or concept maps. This example is fairly straight forward.

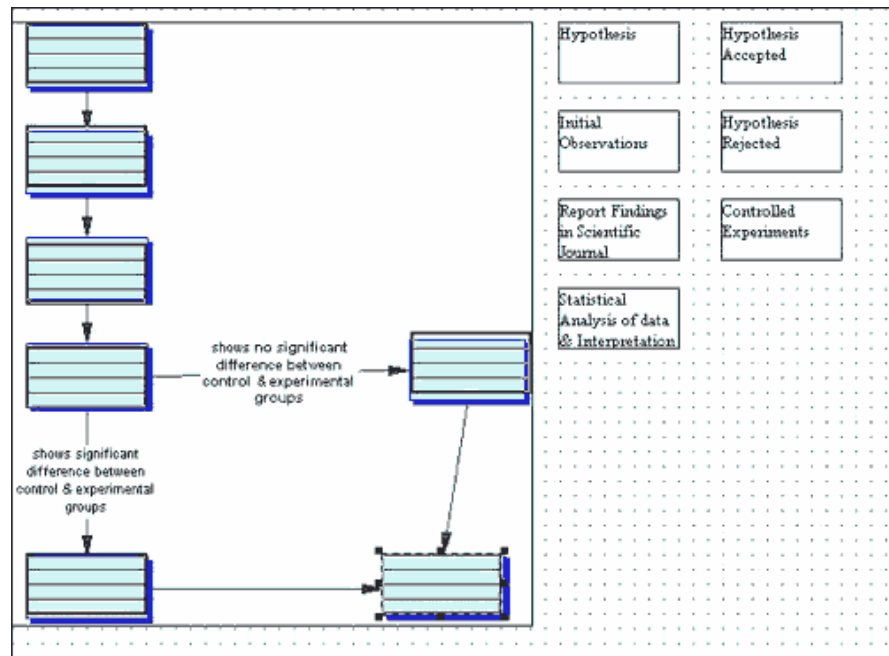
1. Create a new Jamba project that has a width of 600 and a height of 460. These dimensions accommodate the concept map graphic and still play to the lowest common denominator of monitor resolution. Title the concept map "ScientificMethod." Be careful to avoid using spaces in the name because this could lead to problems if this project is placed on a server.
2. Import the concept map graphic using the "add image" button. The graphic is called "scimethod.gif." Place the graphic to the left side of the page.



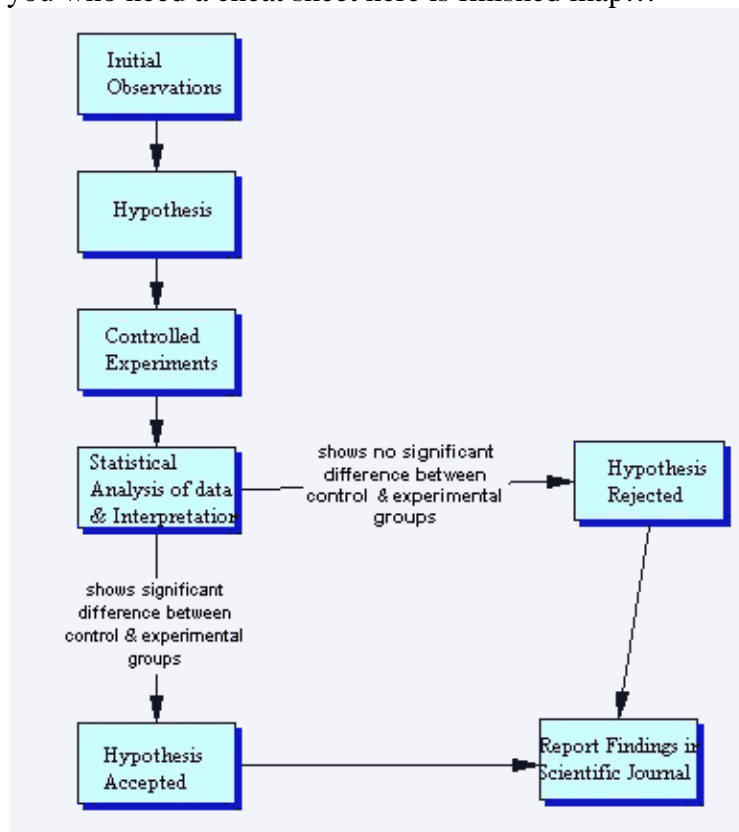
3. Next, create "hot spots" over the empty boxes. Use the hot spot tool from the tool bar to do this.
4. Now click on the "add text" icon on the button bar. Add the following list of items to the canvas. Use a separate text box for each item. For each text-box set the "Transparent" value to true. This option is found in the properties menu. It also helps to change the font to Times New Roman using the properties menu.
  - Controlled Experiments
  - Hypothesis Accepted
  - Report Findings in Scientific Journal
  - Initial Observations
  - Hypothesis Rejected
  - Statistical Analysis of Data and Interpretation
  - Hypothesis

This is what your page should look like.....





- Use the CanBeDraggedTo option in the properties list to set where each text item should be dragged.  
For those of you who need a cheat sheet here is finished map...



That's it. Run this in Netscape and test your knowledge of the scientific method. Amazingly easy to use and, oddly enough, the most effective tool developed, these concept maps earn an effort-to-result rating of an 11 on a scale of 1 to 10. Once you become moderately proficient with Inspiration™ and Jamba™ an average concept map can be created in about 30 minutes.

### Using Animation and the "To Do" List

This example builds on the concepts from the "CAT" example while adding animation and a set of actions called the "To Do" list. The animations that we will be using are generated from image strips. An image strip is much like a movie strip.

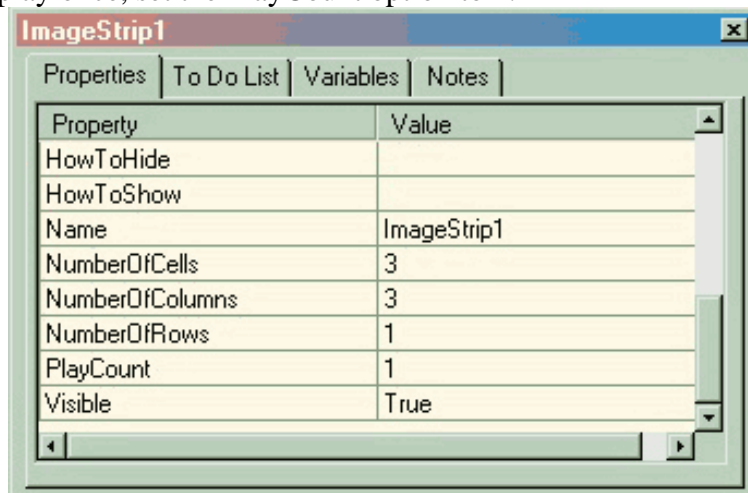


Notice that the strip can be divided into three equal sized "frames." This allows Jamba to slide through the strip creating an animation while keeping the animation in one file. The "To Do" list is a set of events and actions. Events are instances where actions can be triggered. Events are things such as mouse over, mouse movement, and clicking. Events can also be complete or incomplete actions. In the "CAT" example, "drag and drop" was an action. Successfully dragging and dropping was a complete action while dragging but not dropping was an incomplete action. Both of these could potentially trigger events, as will be shown in this example. Movement of materials and tonicity are concepts usually covered in introductory biology courses. They can be difficult concepts, and lend themselves very well to animated demonstrations. In this project we will use an animated red blood cell to simulate the reaction of the cell when placed in beakers of solution with varying solute concentrations. After dragging the cell into the solution and seeing the reaction, the student will be asked questions about the solute concentration of the solution in the beaker.

1. Create a new project called "movement."
2. Click on the "add image" icon on the toolbar. A dialog box will appear. Browse for a file called "beaker.gif."
3. Click on the "add image strip" icon on the toolbar. A dialog box will appear. Browse for the file called "rbc.gif."

## Creating Web Tutorials

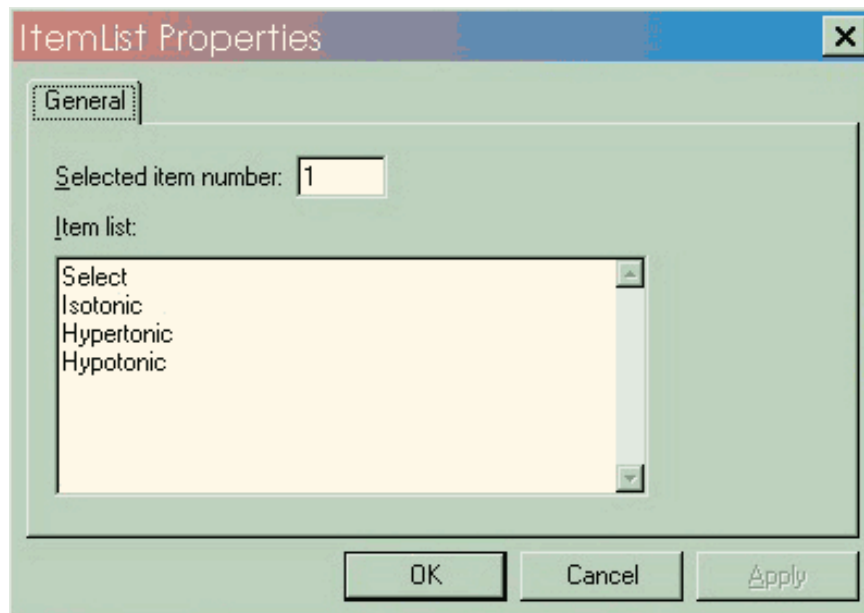
- The image strip is divided into rows and columns, which have to be set in the properties box. The rbc.gif has three columns and one row. The animation only needs to play once, set the PlayCount option to 1.



- Create a hot spot over the lower 2/3 of the beaker. This will be used for a drag drop target and event to play the image strip animation.
- Click on the red blood cell image, then in the properties menu select "CanBeDraggedTo." Then with the targeting tool select the hot spot.
- Switch to the "To Do List." Set the "DragDrop" to play the image strip by selecting "DragDrop" then selecting the image strip with the targeting tool, then set "What to do" to "Play."
- Just above the red blood cell, place text that says "1% Solute." Then to the right side of the canvas, add text that says "Drag the red blood cell into the solution."
- Click on the red blood cell. In the "To Do List" select "Enter," then use the target tool to select the "Drag the red blood cell....." text. In "What to do" select "visible." Then set the "Value" to "false." This will cause the instructional text to disappear.

### Adding questions to the interface

- Click on the combo box icon in the button bar. Drag the box to the top right corner. This box contains the possible answers to a question asked about the animation that happened when the student placed the cell in the beaker.



11. This should not be visible until after the animation has played. Set the "Visible setting" in the properties box to "False."
12. Create text that says "The solution in the beaker is most likely" and another set of text that says "to the red blood cell." Set both of the texts that were just created to visible "false." Line up the text with the combo box.
13. Click on the red blood cell. From the "To Do List" select DragDrop then with the target tool select the combo box. Then set "What to do" to "visible," and the value to "true." Repeat this with both of the pieces of text.
14. Create text that says "Correct." Make the font large and green, and set the visibility to "false." Create another text that says "Try Again." Make this font larger and red and set the visibility to false. Place "Correct" and "Try Again" on top of each other in the middle of the right hand side of the page.
15. Pay very close attention here. Click on the combo box, then switch to the "To Do List." Choose the "Select" option. Next select the "new condition" icon.



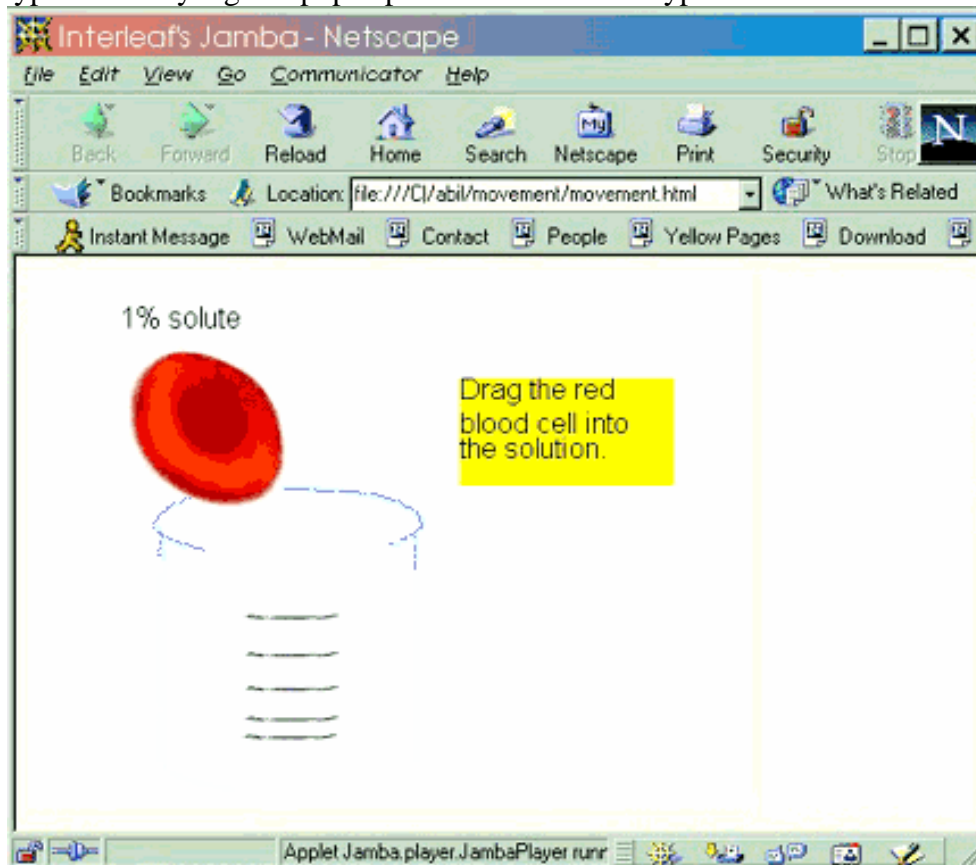
The new condition icon gives an interface to create custom events and actions.

15. In the "IF" box use the target tool to select the combo box. From the list of possible interactions pick "SelectedItemData." Then select "IS ' = = equal to'" if it is not already selected. On the next line type "hypertonic."

## Creating Web Tutorials

16. Click on "Put into Condition." Then click "OK." Use the target tool in the current "To Do" list to select "Correct." Set "what to do" to "visible" "true."
17. Again click on the "new condition" icon. This time create the two incorrect answers. In the "IF" box use the target tool to select the combo box. From the list of possible interactions pick "SelectedItemData." Then select "IS '=' equal to'" if this is not already selected. On the next line type "hypotonic", then click on "Put into Condition." Then go back and replace "hypotonic" with "isotonic." Click on "OR into Condition." Click "OK."
18. Use the target tool in the current "To Do" list to select "Try Again." Set the "what to do" to "visible" "true."

The applet just created, when run, tells the user to drag the cell into the solution. When the cell is dragged into the solution and cell creates, a statement pops up "The solution in the beaker is most likely (isotonic, hypotonic, hypertonic) to the red blood cell." If the user selects isotonic or hypotonic "Try Again" pops up. If the user selects hypertonic then "Correct" pops up.



There is really only one detail left to mop up. When this applet is run, note that if an incorrect answer is selected before a correct answer or some combination of selections, the word "correct" never appears. This is because the applet has never been told to erase the "correct" and "try again" statements. To accomplish this create a general statement that tells the applet to make the words "correct" and "try again" invisible whenever a new item is selected. This involves creating a new interaction in the "To Do" list for the combo box. It is a select action that is set to "Do Always." This should be enough information to put together the statement. The logic looks like this:

**IF** Combo Box item is **SELECTED** **WHERE** item=DO **ALWAYS** **THEN**  
 "correct" or "try again" visible false

### Adding the Applet to an HTML Page:

To add an applet to a web page use the <applet> tag. An applet can be added anywhere in HTML, but the most acceptable location is between the body tags.

Here is an example of the code to use.

```
<html>
<head><title>Movement of Materials Tutorial</title></head>
<body>
<applet code="Jamba.player.JambaPlayer.class" width=400 height=300>
<param name="fps" value="15">
<param name="filename" value="movement.JTF">
<param name="page" value="StartPage">
<param name="AppletColor" value="192, 192, 192">
<param name="AppletName" value="movement">
</applet>
</body>
</html>
```